

PROBABILISTIC COVERING PROBLEMS

A Thesis
Presented to
The Academic Faculty

by

Feng Qiu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
May 2013

PROBABILISTIC COVERING PROBLEMS

Approved by:

Professor Shabbir Ahmed, Advisor
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Asst. Professor Santanu S. Dey,
Advisor
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Asst. Professor David A. Goldberg
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Professor Ellis Johnson
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Asst. Professor James R. Luedtke
Department of Industrial and Systems
Engineering
University of Wisconsin-Madison

Professor George L. Nemhauser
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Date Approved: February 6, 2013

In the beloved memory of my maternal grandmother ...

ACKNOWLEDGEMENTS

My pursuit of a Ph.D. can be likened to a journey of sailing across the ocean, like the one in the movie “Life of Pi,” in which Pi weathered storms but also sunshine. While Pi made his journey with his tiger friend, Richard Parker, I accomplished my journey with tremendous support and guidance that I received from many. My deepest and most sincere gratitude goes to my advisers, Professors Shabbir Ahmed and Santanu S. Dey. I am eternally grateful for their patience and dedication as I progressed from a novice researcher, coming back to school after years of work in industry, to a seasoned researcher. The image of their sitting together with me in the lounge of the Hilton San Diego Bayfront Hotel in 2009, helping me correct my INFORMS presentation slides, is still as fresh as it was then. Their brilliant insights into research, their relentless enthusiasm for exploring the unknown, and their persistence at solving difficult problems are traits that I strive to emulate. Being their student, learning from them, and working with them have been the most rewarding and joyful part of this journey. I could not be more appreciative of their guidance, motivation, and mentorship.

I also wish to express my gratitude to the other committee members, whose insights have been invaluable along the way: Professors David A. Goldberg, Ellis Johnson, James Luedtke, and George L. Nemhauser. I feel fortunate enough to have collaborated with Professor Laurence Wolsey, whose frank advice and clever suggestions considerably enhanced the quality of my work. I am also grateful to Professors Gary Parker and H. Donald Ratliff for their encouragement and financial support in my difficult time. I would also like to extend my sincere thanks to Professors Xueguang Chen, Zongfen Han, and Leyuan Shi, for their farsighted guidance and

generous support in my early academic pursuits and career.

I have greatly benefited from the outstanding faculty members at Georgia Tech and the courses they taught. I enjoyed Linear Programming, taught by Professor Craig Tovey, Computational Methods, taught by Professor Earl Barnes, Integer Programming, taught by Professor George Nemhauser, and Stochastic Programming, taught by Professor Shabbir Ahmed. I have also enjoyed the fresh farm products brought to school by Professor Ellis Johnson. I would like to thank Ms. Jane Chisholm and Mr. Kurt Belgum of the Language Institute for their help with my English, and Ms. Pamela Morrison and Ms. Valarie DuRant-Modeste for their administrative support.

One of the most precious treasures I have collected during my years in ISYE is my colleagues and friends that I have made here. I have enjoyed stimulating discussions and incredibly fun idea exchanges with Qie He, Akshay Gupte, Dimitri Papageorgiou, Helder Inacio, Jon (Pete) Petersen, Fatma Kilinc Karzan, Steve Tyber, Yaxian Li, Xuefeng Gao, Linwei Xin, Minghui Yu, Li Xu, and Murat Yildirim. I thank Fei Qian, Antonio Carbajal, Adam Esen, Kathleen Lindsey, Jackie Griffin, Kan Wu and Vinod Cherrian, for their encouragement and counseling and James Luedtke and Yeliz Ekinici for being wonderful officemates. I would also like to thank Diego Moran Ramirez, Rodolfo Carvajal, and Gustavo Angulo for granting me membership into the ISYE Chilean “mafia,” and Işıl Alev and Bahar Çavdar, for introducing me to the Turkish community. I would like to extend my sincere appreciation to all other friends I have made during this journey—to just name a few, Shan Ba, Jason Cullen, Shuhua Dai, Niao He, Md. Moinul (Moin) Islam, Ran Jin, Guanghui Lan, Dexin Luo, Aly Samy Megahed, Liang Pi, Norbert Remenyi, Yingzhe Shi, Dan Steffy, Peng Tang, Carlos Felipe Valencia, Huizhu Wang, Peng Wang, Qianyi Wang, Shuangjun Xia, Fangfang Xiao, Yi Xiao, Hoksung Yau, Haiyue Yu, Jiangchuan (Ralph) Yuan, Hao (Haward) Zhang, Weidong Zhang, Yang Zhang, Yu Zhang—all of whom have made this journey

more enjoyable. I also deeply appreciate the unwavering support from my old friends —Ying Chen, Chen Li, Jianhua Peng, Yipeng Wang, and Chuan Zhang—even we were thousands of miles apart.

I am extremely thankful to my wife, Jingjing Gao, and my parents, Xianguo Qiu and Huanwen Fu, for their steadfast support during all of my endeavors. I would like to say thank them from the bottom of my heart for being my bedrock of love and support, without which I could not have come this far.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	x
LIST OF FIGURES	xi
SUMMARY	xii
I INTRODUCTION	1
1.1 Probabilistically Constrained Programs	1
1.1.1 Optimization Under Uncertainty and Probabilistic Constraint	1
1.1.2 Probabilistically Constrained Programs	3
1.1.3 Probabilistic Covering Problems	6
1.2 Mixed-Integer Programs	7
1.2.1 Branch-and-Bound	8
1.2.2 Valid Inequalities (Cutting Planes)	10
1.3 Dissertation Overview	12
1.3.1 Probabilistic Covering Linear Programs	12
1.3.2 Probabilistic k -Cover Problems	14
II PROBABILISTIC COVERING LINEAR PROGRAMS	17
2.1 Introduction	17
2.2 Difficulty of Solving CKVLP	20
2.2.1 Computational Complexity	20
2.2.2 Performance of a standard MIP solver on CKVLP instances	25
2.3 Iterative Coefficient Strengthening	28
2.4 Mixing Set Inequalities	34
2.4.1 The basic mixing closure	36
2.4.2 Bound Quality	39
2.5 Branching Scheme	40

2.6	Computational Experiments	41
2.6.1	Implementation Details	42
2.6.2	Probabilistic Portfolio Optimization	43
2.6.3	Optimal Vaccination Allocation	47
2.7	Concluding Remarks	54
III	CUTTING PLANES FOR PROBABILISTIC COVERING LINEAR PROGRAMS	56
3.1	Introduction	56
3.2	Valid Inequalities Induced by Resource Constraints	57
3.3	Valid Inequalities From Two Row Constraints	62
3.4	Computational Experiments	70
IV	PROBABILISTIC K-COVER PROBLEMS	73
4.1	Introduction	73
4.2	Ambiguous Probabilistic k -Cover Set	77
4.2.1	Definition	77
4.2.2	Deterministic Reformulation	78
4.3	Fully Aggregated Model (FAM)	82
4.3.1	Inner and Outer Approximations	84
4.3.2	MIP Formulations	84
4.3.3	Numerical Examples	89
4.4	Partially Aggregated Model (PAM)	94
4.4.1	Partially Aggregated Model	94
4.4.2	Strengthened PAM	98
4.4.3	Gaps Between PAM and FAM	102
4.4.4	Mixed-Integer Formulations	110
4.4.5	Bounds for Dual Extreme Points	113
4.4.6	Numerical Examples	120
4.5	Applications of Modeling Techniques	123
4.5.1	Discrete Distributions With Non-uniform Upper Bounds	123

4.5.2	Location Problems	124
4.5.3	Probabilistic Shortest Path Problems	126
4.6	Conclusions	129
REFERENCES		130

LIST OF TABLES

1	Performance of CPLEX on CKVLP	26
2	Percentage Improvements Over CPLEX (Portfolio Optimization with Budget Constraint)	45
3	Percentage Improvements Over CPLEX (Portfolio Optimization Without Budget Constraint)	45
4	Computational Results (Optimal Vaccination Allocation Problem) . .	50
5	Percentage Improvements Over CPLEX (Optimal Vaccination Allocation Problem)	51
6	Comparison of Performance on Solving a MIPLIB Instance	72
7	Probability Distributions in Example 1	105
8	Probability Distributions in Example 2	106
9	Probability Distributions in Example 3	106
10	Comparison of Lower Bounds	106
11	Comparison of Lower Bounds ($n=10$)	107
12	Comparison of Lower Bounds ($n=10$) (Independent Distributions) . .	107
13	Probability Distributions in Example 4	108
14	Probability Distributions in Example 5	108
15	Probability Distributions in Example 6	109
16	Comparison With Results in [100]	109
17	Lower Bounds for 3-Coverage	110

LIST OF FIGURES

1	LP Bound Improvement Over Time/Iteration ($m = 1000$)	53
2	Bounds for 3-Coverage Single Target Instances ($n=10$)	91
3	Solution Reliability with Single Target Instances	93
4	Comparison of Bounds by FAM and SPAM	122

SUMMARY

This dissertation studies optimization problems that involve probabilistic covering constraints. A probabilistic constraint evaluates and requires that the probability that a set of constraints involving random coefficients with known distributions hold satisfy a minimum requirement. A covering constraint involves a linear inequality on non-negative variables with a greater or equal to sign and non-negative coefficients. A variety of applications, such as set cover problems, node/edge cover problems, crew scheduling, production planning, facility location, and machine learning, in uncertain settings involve probabilistic covering constraints.

In the first part of this dissertation we consider probabilistic covering linear programs. Using the sampling average approximation (SAA) framework, a probabilistic covering linear program can be approximated by a covering k -violation linear program (CKVLP), a deterministic covering linear program in which at most k constraints are allowed to be violated. We show that CKVLP is strongly NP-hard. Then, to improve the performance of standard mixed-integer programming (MIP) based schemes for CKVLP, we (i) introduce and analyze a coefficient strengthening scheme, (ii) adapt and analyze an existing cutting plane technique, and (iii) present a branching technique. Through computational experiments, we empirically verify that these techniques are significantly effective in improving solution times over the CPLEX MIP solver. In particular, we observe that the proposed schemes can cut down solution times from as much as six days to under four hours in some instances. We also developed valid inequalities arising from two subsets of the constraints in the original formulation. When incorporating them with a modified coefficient strengthening procedure, we are able to solve a difficult probabilistic portfolio optimization instance

listed in MIPLIB 2010, which cannot be solved by existing approaches.

In the second part of this dissertation we study a class of probabilistic 0-1 covering problems, namely probabilistic k -cover problems. A probabilistic k -cover problem is a stochastic version of a set k -cover problem, which is to seek a collection of subsets with a minimal cost whose union covers each element in the set at least k times. In a stochastic setting, the coefficients of the covering constraints are modeled as Bernoulli random variables, and the probabilistic constraint imposes a minimal requirement on the probability of k -coverage. To account for absence of full distributional information, we define a general ambiguous k -cover set, which is “distributionally-robust.” Using a classical linear program (called the Boolean LP) to compute the probability of events, we develop an exact deterministic reformulation to this ambiguous k -cover problem. However, since the boolean model consists of exponential number of auxiliary variables, and hence not useful in practice, we use two linear program based bounds on the probability that at least k events occur, which can be obtained by aggregating the variables and constraints of the Boolean model, to develop tractable deterministic approximations to the ambiguous k -cover set. We derive new valid inequalities that can be used to strengthen the linear programming based lower bounds. Numerical results show that these new inequalities significantly improve the probability bounds. To use standard MIP solvers, we linearize the multi-linear terms in the approximations and develop mixed-integer linear programming formulations. We conduct computational experiments to demonstrate the quality of the deterministic reformulations in terms of cost effectiveness and solution robustness. To demonstrate the usefulness of the modeling technique developed for probabilistic k -cover problems, we formulate a number of problems that have up till now only been studied under data independence assumption and we also introduce a new applications that can be modeled using the probabilistic k -cover model.

CHAPTER I

INTRODUCTION

This dissertation studies two types of optimization problems that involve probabilistic constraints: a class of probabilistic covering linear programs, and a class of probabilistic binary integer programs. The general theme of our study is to develop deterministic mixed integer linear programming (MILP) based solution approaches for these stochastic problems. In this chapter, we introduce probabilistically constrained optimization problems, review some general techniques for modeling and solving MILPs that are relevant to our research, and present an outline of this dissertation.

1.1 Probabilistically Constrained Programs

In this section, we first present an overview of optimization under uncertainty and the motivations of using a probabilistic constraint to address uncertainties in optimization problems. Then we review the challenges of solving probabilistically constrained programs and possible solution approaches. Finally, we introduce the topic of this dissertation: probabilistic covering problems.

1.1.1 Optimization Under Uncertainty and Probabilistic Constraint

Optimization (alternatively, mathematical optimization or mathematical programming) is the analysis and solution of a problem in which a best choice (with regard to the decision maker's criteria) is selected from a set of “feasible” choices. Consider the following optimization problem,

$$\min(\max) : f(x, u)$$

$$G(x, w) \geq 0,$$

where $x \in \mathbb{R}^n$ are decision variables, u and w are input data (or parameters) assumed to be deterministic; $G(x, w) \geq 0$, where $G(x, w) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, defines all feasible choices; and $f(x, u) : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, representing the decision maker's criteria for being a best choice. The maximization or minimization operator seeks a best decision with regard to the decision maker's criteria.

In the past several decades, optimization theory and applications have undergone a significant development. However, one criticism of deterministic optimization is that its solution is not robust (e.g., the optimal choice may not be the best choice or even not feasible in some future scenarios) in a real application setting where uncertainties are present in input data (e.g., u , w , or both). When u is uncertain, $f(x, u)$ becomes a random number; when w is uncertain, $G(x, w)$ is a random vector, and whether $G(x, w) \geq 0$ holds becomes a random event. Although sensitivity analysis or perturbation analysis provides some local views on the consequences of the uncertainties, traditional deterministic optimization lacks tools that can handle uncertainties explicitly. To systematically address optimization problems with stochastic data, we often need to incorporate statistical measures into models (e.g., we take the expectation of $f(x, u)$; we evaluate the probability that $G(x, u) \geq 0$.)

Depending on the perspective of a decision maker and the nature of uncertainties, different statistical measures can be employed. If the decision maker emphasizes “average performance,” we optimize the expectation (or mean) of the objective function $f(x, u)$, i.e., $\mathbb{E}(f(x, u))$; if the decision maker also concerns the risk of the objective value falling out of an acceptable range, the combination of mean and variance can be considered as a measure, i.e., $\mathbb{E}(f(x, u)) + c\mathbb{D}(f(x, u))$, where \mathbb{D} represents the variance and c is a weight for the variance assigned by the decision maker [4]; if the feasibility of a decision is the major concern, then the probability that $G(x, u) \geq 0$ needs to be evaluated and constrained, given that the distributions of w are known. Therefore, the *probabilistic constraint* (or *chance constraint*) is a constraint that requires that

the probability that a set of constraints is satisfied meet a minimal requirement $1 - \epsilon$, where ϵ is a risk level selected by the decision maker, i.e.,

$$\mathbb{P}\{G(x, w) \geq 0\} \geq 1 - \epsilon, \quad (1)$$

where $\mathbb{P}\{E\}$ evaluates the probability of event E .

Probabilistically constrained programs are often compared with robust optimization, in which each solution must satisfy $G(x, w) \geq 0$ for every possible value of w (every scenario), and considered a relaxed version of robust optimization. One drawback of robust optimization is that it immunizes against for every possible scenario, including the worst, which may require extremely-high costs but which has a very slim chance of occurring. In many realistic settings, decision makers prepare for the scenarios that are most likely to occur, not every possible scenario. Therefore, robust optimization might not be an economical tool for decision making in which an absolutely feasible solution is not necessary. A way of avoiding the selection of solutions with high cost but little likelihood is to assign probabilities to every scenario and to require that the constraints be satisfied “most of the time.” This is one of the most important motivations for probabilistically constrained programs.

1.1.2 Probabilistically Constrained Programs

Optimization problems subject to one or more probabilistic constraints are called *probabilistically constrained programs* (or *chance constrained programs*), i.e.,

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \mathbb{P}\{G(x, \zeta) \geq 0\} \geq 1 - \epsilon \\ & x \in X, \end{aligned} \quad (2)$$

where ζ is random data with known distributions; and X is some set defined by deterministic constraints. Note that in (2) the probabilistic constraint is applied on

a set of constraints; we can also apply the probabilistic constraint to each of the constraints individually.

Probabilistically constrained programming was first introduced in 1959 [34] by Charnes and Cooper. Since then, it has been applied to a broad range of problems that involve uncertain data, such as electrical power generation, reservoir operations, inventory management, and facility planning [102]. However, the solution of probabilistically constrained problems remains computationally challenging for the following two reasons:

- Evaluation of the probability that a set of constraints holds given a vector x , which requires high-dimensional integration, is extremely difficult. Even for the single row case, the evaluation may not be possible for general distribution functions.
- The probabilistically constrained set, i.e., $\{x : \mathbb{P}\{G(x, w) \geq 0\} \geq 1 - \epsilon\}$, is not convex in general, imposing severe challenges for the purposes of optimization.

Various solution approaches have been proposed with regard to the nature of the underlying uncertainties and the structure of the constraints. In some special cases in which function G and distribution functions of w satisfy certain properties, the feasible regions are convex [104, 33, 97, 30]. When G is a linear function, and uncertainties with finite supports appear on only the right-hand sides of the linear inequalities, the concept of “ p -efficient points” is introduced to obtain strong MIP reformulations that characterize the feasible regions [47, 46]. However, because probabilistically constrained programs with general probability distributions do not have the convex property, they still remain difficult to solve. Recently, a number of studies have solved tractable conservative convex approximation to obtain feasible solutions [17, 22, 28, 53, 94, 51]. However, the solutions are often highly conservative and lack guarantee about the quality of approximation.

Another type of approximation is the sample-average approximation (or SAA), in which N samples (scenarios) are drawn from the distribution and the frequency of the scenarios in which constraints are satisfied is used to approximate the probability that the constraints hold. It has been shown that when the number of samples, N , is large enough, the solution by SAA has a high probability of being a feasible solution to the original problem [81].

One advantage of the sample-average approximation (or SAA) is that it can approximate any chance constrained problems as long as the underlying distribution can be sampled because the evaluation of probability is replaced by counting the number of scenarios in which the constraints are satisfied. Furthermore, if the original constraints $G(x, w)$ are linear, the resulting formulation will be a mixed-integer linear program that can be solved by standard MIP solvers. SAA also has several drawbacks:

- The feasibility of the solutions (with regard to the original problem) by SAA is probabilistic, requiring posterior validation.
- To obtain a feasible solution or a valid bound with a high confidence level, a large number of samples are required, resulting in a large-scale mixed integer program.

In the first part of this dissertation, we develop efficient approaches to solve the MIP formulations of sample average approximations of probabilistic covering linear programs; In the second part of this dissertation, we develop deterministic approximations for probabilistic k -cover problems that provide deterministic bounds but require no sampling or posterior validation.

1.1.3 Probabilistic Covering Problems

A covering inequality is a linear inequality on non-negative variables with a “greater or equal to” sign and non-negative coefficients:

$$\sum_{j=1}^n a_j x_j \geq b,$$

where $a_j \geq 0$, $j = 1, \dots, n$ and x_j s are non-negative decision variables. The right-hand side b is generally strictly positive; otherwise, the covering inequality is trivial. Covering inequalities are widely used constraints arising in many mathematical programming problems, such as set cover problems, node/edge cover problems, crew scheduling, production planning, facility location, and machine learning, to name just a few.

A covering problem is a mathematical programming problem that has covering inequalities as the major structure:

$$\begin{aligned} \min \quad & \sum_{j=1}^n c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = 1, \dots, m \\ & x \in X \\ & x_j \geq 0 \quad j = 1, \dots, n, \end{aligned} \tag{3}$$

where X is some set defined by additional constraints, e.g., integrality constraints. Since the covering inequalities restrict the feasible region to be in an up-hull, objective coefficient vector c is normally non-negative. A covering problem seeks a vector with minimal cost whose non-negative linear combinations cover the right-hand sides. When the set X in (3) is given by simple bounds, then (3) is a covering linear program; when $X = \{0, 1\}^n$, $a_{ij} \in \{0, 1\}$ for all i and j , and $b_i = 1$ for all i , then (3) is a set covering problem. If b_i are allowed to be larger than one in the set covering problem, then (3) generalizes the classic set cover problem and becomes a set multi-cover

problem.

Uncertainties in data lead to the stochastic counterparts of covering problems. For example, in a deterministic set covering problem, we seek a set of subsets whose union constitutes the whole ground set. Given a set of subsets, we can immediately determine if these subsets cover the ground set. Consider a stochastic setting where an element belongs to a subset with certain probability, i.e., $\mathbb{P}\{a_{ij} = 1\} = p_{ij}$ and $0 < p_{ij} < 1$. Given a set of subsets, the validity of the covering inequality is not deterministic any more and we can only require that the element be covered by the chosen subsets with a certain probability. We consider a probabilistically constrained (or chance-constrained) optimization approach to address covering problems with uncertain data. A probabilistic covering problem is as follows:

$$\begin{aligned}
& \min \sum_{j=1}^n c_j x_j \\
& \text{s.t. } \mathbb{P}\left\{\sum_{j=1}^n \tilde{a}_{ij} x_j \geq \tilde{b}_i\right\} \geq 1 - \epsilon_i \quad i = 1, \dots, m \\
& x \in X \\
& x_j \geq 0 \quad j = 1, \dots, n,
\end{aligned} \tag{4}$$

where \tilde{a}_{ij} and \tilde{b}_i are non-negative random numbers following some known distributions for all i and j . The i -th probabilistic constraint enforces the probability that the right-hand side is covered is at least $1 - \epsilon_i$, which is called the “reliability level” (risk level). The probabilistic constraint can also be applied jointly for all constraints.

1.2 *Mixed-Integer Programs*

In this section, we review the mixed-integer program technologies that are relevant to our studies. A mixed-integer program is a mathematical program in which some of the variables are restricted to be integers. In our context, we also restrict our attention to mixed-integer linear programs, in which the constraints and objective functions are linear functions, i.e.,

$$\begin{aligned}
& \min cx + dy \\
& \text{s.t. } Ax + Gy \geq b \\
& x \in \mathbb{R}_+^n, y \in \mathbb{Z}_+^p,
\end{aligned} \tag{5}$$

where \mathbb{R}_+^n is the set of nonnegative real n -dimensional vectors and \mathbb{Z}_+^p is the set of nonnegative integral p -dimensional vectors; x and y are decision variables; A and G are rational matrices with appropriate dimensions. The feasible set is defined as $S = \{(x, y) \in \mathbb{R}_+^n \times \mathbb{Z}_+^p : Ax + Gy \geq b\}$. We refer interested readers to [93] for a comprehensive discussion about the concepts and algorithms for mixed-integer linear programs.

1.2.1 Branch-and-Bound

Branch-and-Bound, a successful algorithm for solving mixed-integer linear programs globally and optimally, has been implemented as the generic framework for solving mixed-integer programs in every commercial solver. Because of the presence of integer variables y in the problem, the overall feasible region is naturally a union of a finite number of subregions, each of which is defined by fixing some of the integer variables y at integral values. Branch-and-bound partitions the feasible region progressively into smaller subregions and conducts a search in each subregion. The partitions (or nodes) of the feasible region are organized in a tree structure, also known as a “branch-and-bound tree”.

For a mixed-integer linear program, branching can be done by specifying the constraints on integer variables. For example, let k be a positive integer and y_i be one of the integer variables, then a branching on y_i can be done by taking $S = S_1 \cup S_2$, where $S_1 := S \cap \{y_i \leq k\}$ and $S_2 := S \cap \{y_i \geq k + 1\}$ [93]. S_1 and S_2 are called “subproblems” or “node problems”. Another option is to branch on constraints. In

commercial mixed-integer program solvers, only dichotomy branching is implemented and binary trees are used.

To avoid complete enumeration, a MIP solver uses a pruning mechanism to remove the nodes that do not have an optimal solution: During the search process, the best feasible solutions so far (called the “incumbent solution”) is kept. It provides an upper bound, z^{UB} , on the optimal objective function value z^* . If the lower bound of the objective function value, z^{LB} , at a node is no smaller than z^{UB} , then this node will not yield a feasible solution with a better optimal objective value than the incumbent solution, and hence, it can be pruned. A node will also be pruned if the subproblem is identified as an infeasible problem. (We will develop such an infeasibility-based pruning technique in Chapter 2.) When each of the nodes is either explored or pruned, branch-and-bound terminates with an optimal solution or no solution at all.

A straightforward lower bound can be obtained by solving the linear program relaxation of the mixed-integer linear program. For example, the linear program relaxation for the root node problem, in which no branching has been done yet, is as follows:

$$\begin{aligned} \min \quad & cx + dy \\ \text{s.t.} \quad & Ax + Gy \geq b \\ & x \in \mathbb{R}_+^n, y \in \mathbb{R}_+^p. \end{aligned} \tag{6}$$

Note that the integrality constraints have been dropped. Since any feasible solution to (5) is also a feasible solution to (6), the optimal objective function value of (6) provides a lower bound for the optimal objective function value of (5). Clearly, we could obtain a tighter lower bound by utilizing the integrality constraints on y , which, in turn, helps the branch-and-bound process, discussed in the following section. An upper bound can be obtained if the solution of the linear program relaxation

is integral.

In addition to lower and upper bounding, we may also conduct node selection and branching variable selection to expedite the branch-and-bound process. Node selection, given a set of nodes (subproblems), is choosing a node that will be examined next. In addition to using the two common strategies, depth-first search and breadth-first search, we may also follow practical rules such as “best lower bound,” choosing the node with the best lower bound, and “best estimate,” choosing the node likely to contain an optimal solution. Branching variable selection is the decision to choose an integer variable to branch on in a node subproblem. One can either specify higher priorities on some decision variables according to his/her prior knowledge about the optimization problem being solved or perform tentative branching on candidate variables and choose the variable that can cause the largest increase in the objective function value [93].

1.2.2 Valid Inequalities (Cutting Planes)

For branch-and-bound to terminate with a provable optimal solution in a timely manner, lower bounds with good quality are important. Since the branch-and-bound process is very dependent on LP-relaxation of the original formulations, formulations with tight LP-relaxation are a good starting point. In Chapter 2, we introduce a coefficient strengthening technique that tightens the original formulations.

Another important technique for improving the lower bound for mixed-integer linear programs is to add valid inequalities (or cutting planes). A mixed-integer linear program minimizes (or maximizes) a linear function over its mixed-integer set S , so an optimal solution can be attained on the boundary of the convex hull of S , which is the convex combination of all points in S . The convex hull of a mixed-integer set with rational data has been shown to be a polyhedron. Thus, ideally, if we know all the inequalities defining the convex hull of S , we can solve the original mixed-integer

linear program as a linear program. However, in most practical situations this is not possible, either because of too many such inequalities or lack of knowledge about their forms. A less ambitious approach is to add some linear inequalities that are valid for all points in set S but that can also cut off some part of the feasible region of the LP-relaxation of S with the purpose of improving the lower bounds and expediting the branch-and-bound process. Such a linear inequality is called a “cutting plane” or a “valid inequality.”

Some valid inequalities are general and can be applied to most integer programs such as Gomory mixed-integer inequalities, mixed-integer rounding inequalities, disjunctive inequalities, intersection inequalities. Other types of valid inequalities are problem-specific inequalities, which are more closely related to the work presented in this dissertation. In Chapter 2, we discuss a family of valid inequalities, mixing set inequalities, arising from the k -violation structure of probabilistic covering linear programs; in Chapter 3, we discuss two families of valid inequalities arising from two relaxations of the original MIP formulation.

Valid inequalities can be added to the root node problems, which are LP-relaxations of the original problems, or to node subproblems during the branch-and-bound process, called “branch-and-cut.”

Now we briefly introduce two techniques for generating the valid inequalities used in this dissertation.

- **Lift-and-project:** Some mixed-integer sets can be lifted into sets that reside in a higher dimensional space using auxiliary variables and that are defined by linear constraints. The projections of the sets in higher dimensional space back to the original space yield valid inequalities for the original mixed-integer sets. Lift-and-project can be used to derive valid inequalities prior to computation or to separate violated inequalities in runtime. We use this technique to generate the first class of valid inequalities in Chapter 3.

- **Lifting:** Lifting creates a valid inequality for the original set, given one for a lower dimensional set, by augmenting a binary variable that is absent from the original inequality [70]. We generate the second family of inequalities in Chapter 3 using this technique.

1.3 Dissertation Overview

In this thesis we study (i) MIP methods for solving sample average approximations of covering linear programs and (ii) deterministic MIP approximations of probabilistic k -cover problems. Next we provide a brief overview of the thesis.

1.3.1 Probabilistic Covering Linear Programs

In Chapter 2 we study sample average approximation of probabilistic covering linear programs. The SAA reformulation structure resembles a k -violation linear program, a linear program that allows at most k constraints to be violated, because SAA approximates the probabilistic constraint by allowing the violation of covering constraints in at most k scenarios. We first study the computational complexity of solving covering k -violation linear programs (or CKVLP) and show that it is strongly NP-hard by reducing from vertex cover problems. We also empirically examine the factors that affect the computational time when solving a CKVLP.

To improve the performance of mixed-integer programming (MIP) based schemes for these problems, we introduce and analyze a coefficient strengthening scheme, adapt and analyze an existing cutting plane technique, and present a branching technique. The coefficient strengthening technique tightens the coefficient of a binary variable using a lower bound for the left-hand side of the corresponding covering linear inequality by solving a linear program. The technique can be applied iteratively. We show that the iterative strengthening procedure terminates in a finite number of iterations and present an upper bound on the maximal gap that could possibly be closed by

the procedure. Then, we use mixing set inequalities, which have been previously proposed to solve probabilistically constrained problems, to strengthen the lower bound by LP-relaxation. The procedure first constructs a relaxation, which is in the form of mixing set, of the original problem using the k -violation substructure and then generates valid inequalities for this relaxation. Since there are infinitely many mixing set relaxations, and hence infinitely many mixing set inequalities, the maximal gap that can be closed by this procedure is determined by the closure formed by all the mixing set inequalities. We point out that the closure is contained in another closure that is a polyhedron, and we use the latter closure to derive an upper bound for the maximal gap that could be closed by the mixing set inequalities. We introduce a branching rule, which removes the overlaps in the branch-and-bound tree by adding simple cuts.

We test the performance of the proposed approaches on two classes of instances. The first class is a portfolio selection problem, in which a unit of investment is distributed among n assets and the overall return needs to achieve a minimal target. Since each asset has an uncertain return, we model this problem with probabilistic constraints and require the overall targeted return be achieved with a high probability. The second class of test instances is an optimal vaccination allocation problem under uncertainty. In this application, a scarce vaccine is allocated to households in a community to prevent an epidemic from breaking out by restricting the post-vaccination reproductive number to be strictly less than one. Computational experiments on the two classes of problems show that the proposed methods are effective in significantly reducing running time. The coefficient strengthening is most effective for large instances and reduces the solution time and the number of search tree nodes by 80% to 98% in these instances. The branching scheme reduces the size of search trees by removing overlaps between branches and incurring infeasibility-based node pruning. It takes no effort to implement and works most effectively on the CKVLP models

with side constraints. The mixing set cuts are capable of closing a large percentage of root node gaps. However, the impact of these cuts on the branch-and-bound process is mixed. Perhaps better performance might be achieved by a more effective separation procedure for mixing inequalities. We have also investigated the performance of various combinations of the three schemes, but the gains are not significant.

In the third chapter, we develop problem-specific cutting planes for solving the MILP formulation resulting from the sample average approximations of the probabilistic covering linear programs. The cutting planes developed in this chapter arise from the relaxation formed by a subset of constraints of the original MIP formulation. We use lift-and-project technique to derive the first family of valid inequalities and lifting for the second family of valid inequalities. In the implementation, we incorporate the two families of inequalities into a modified version of the coefficient strengthening technique. We conduct computational experiments on a portfolio instance, introduced in Chapter 2, of a larger size, which is taken in the list of MIPLIB 2010. Using the solution approach developed in this chapter, we are able to solve this instance that cannot be solved solely by the approaches proposed in Chapter 2.

1.3.2 Probabilistic k -Cover Problems

In the fourth chapter, we focus our attention on probabilistic k -cover problems, which is a stochastic version of the set k -cover problems. A set k -cover problem is to seek a collection of subsets with a minimal cost whose union covers each element in the set at least k times. The set k -cover problem has many applications. In sensor network deployment, a target is often required to be monitored or detected simultaneously by more than one sensor. In ambulance coverage problems, two or more ambulances within a certain range are assigned to each patient for backup coverage. In a realistic setting, e.g., in the sensor deployment problem, whether a sensor can detect a target or not is affected by many uncertain factors and is a random event. We use a Bernoulli

random number to model whether a sensor can detect a certain target and use a probabilistic constraint to model the requirement on the reliability of k -coverage. Then, we develop deterministic approximations for the probabilistic k -cover problems. Our approaches distinguish from previous studies in the following ways:

1. We do not assume data independence across columns.
2. We study a general case where k can be more than one and the left-hand side is random.
3. Our approximations are deterministic, producing bounds with 100% confidence level and requiring no probabilistic validation.

In practice, the distributions of the Bernoulli random data are often incomplete, e.g., only moments up to level m are known, unless the Bernoulli random numbers are independent from each other. We define an ambiguous k -cover set, which is a “distributionally robust” model in the sense that any point in the ambiguous k -cover set is feasible under all distributions that exhibit the known information. Then, we use the Boolean LP model [25, 66] to evaluate the probability of k -coverage and develop an exact deterministic reformulation. However, since the Boolean model consists of an exponential number of auxiliary variables, it is not useful in practice. Then we use two linear program based bounds on the probability that at least k events occur, which can be obtained by aggregating the variables and constraints in the Boolean model, to develop tractable deterministic approximations to the ambiguous k -cover set. Since the quality of the approximations is very dependent on the probability bounds, we derive extra constraints that can be used to strengthen the linear programming based lower bounds. Numerical results show that these extra constraints significantly improve the probability bounds. To use standard MIP solvers, we linearize the multi-linear terms in the approximations and develop mixed-integer linear programming formulations. We conduct computational experiments to demonstrate

the quality of the deterministic reformulations in terms of cost-effectiveness and solution robustness. To demonstrate the usefulness of modeling technique developed for probabilistic k -cover problems, we formulate a number of problems that have up till now only been studied under data independence assumption, and we also introduce a new application that can be modeled using the probabilistic k -cover model.

CHAPTER II

PROBABILISTIC COVERING LINEAR PROGRAMS

2.1 *Introduction*

In this chapter, we study probabilistic covering linear programs. Covering linear programs are linear programs that have covering inequalities as major constraints. They appear in many application models such as transportation problems, production planning problems, supply management, intensity-modulated radiation therapy (IMRT), resource allocation problems. The covering linear programs are also closely related to packing linear programs where the major constraints are packing inequalities. When the coefficients of the covering inequalities are random, whether the right-hand sides are covered or not becomes a random event. The probabilistic constraints on the covering inequalities evaluate the probability of the coverage and require this probability to be at least $1 - \epsilon$. We study the general case where random data with general distribution functions appear on the left-hand sides of the linear inequalities. We use sampling average approximation (SAA) to reformulate the probabilistic covering linear programs as mixed-integer programs, and then we focus on MIP approaches.

Because SAA approximates the probabilistic constraint by allowing the violation of covering constraints in at most k scenarios, its structure resembles a k -violation linear program (or KVLP), a linear program that allows at most k constraints to be violated[115]:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x \geq b_i \quad i = 1, \dots, m, \\ & \text{at most } k \text{ of the } m \text{ constraints can be violated,} \\ & x \in \mathbb{R}_+^n. \end{aligned} \tag{7}$$

The feasible region of a KVLP is the union of $\binom{m}{k}$ polyhedral sets, each of which are defined by the intersection of some subset of $(m - k)$ inequalities from the m inequalities in (7). In general, such a feasible region is nonconvex and KVLP is a strongly \mathcal{NP} -hard optimization problem [9]. Much of the existing work on this class of problems focuses on polynomial time algorithms for low dimensional problems (i.e. n is fixed and small) (cf. [32] for a survey).

In our context, a_i and b_i are non-negative for all i . We call such a problem a *covering-type k -violation linear program* (CKVLP). CKVLPs, which are an important subclass of KVLPs, have many applications. As a concrete example, consider a probabilistically-constrained portfolio optimization problem [95] to determine a minimum cost allocation x of a unit investment among n assets with uncertain returns, requiring the overall return to be at least r with a probability of $1 - \epsilon$, where $\epsilon \in (0, 1)$ is a pre-specified risk level. A formulation of this problem is

$$\begin{aligned}
\min \quad & c^\top x \\
\text{s.t.} \quad & e^\top x = 1 \\
& \mathbb{P}\{\tilde{a}^\top x \geq r\} \geq 1 - \epsilon \\
& x \in \mathbb{R}_+^n,
\end{aligned} \tag{8}$$

where \tilde{a} is the random return vector for n assets following some known distribution, $\mathbb{P}\{A\}$ denotes the probability of the random event A , c is the cost vector, and $e \in \mathbb{R}^n$ is a vector of ones. A common approach to dealing with the probabilistic constraint in (8) is the sample average approximation method [81] where the distribution of \tilde{a} is approximated by an empirical distribution corresponding to an i.i.d sample of return

vectors $\{a_i\}_{i=1}^m$. The approximated problem then reads as follows:

$$\begin{aligned}
\min \quad & c^\top x \\
\text{s.t.} \quad & e^\top x = 1 \\
& a_i^\top x \geq r \quad i = 1, \dots, m, \\
& \text{at most } k \text{ of the } m \text{ covering inequalities can be violated,} \\
& x \in \mathbb{R}_+^n,
\end{aligned} \tag{9}$$

where $k = \lfloor m\epsilon \rfloor$. Since the return is non-negative and only nonnegative investments are allowed, (9) is an example of CKVLP with an additional equality constraint. In Section 2.6, we discuss a similar application of CKVLP in an optimal vaccine allocation under probabilistic constraints [114]. Additional applications of CKVLP arise in intensity modulated radiation therapy (IMRT) planning [116] and signal broadcasting coverage design [103].

A CKVLP can be modeled as a mixed integer program (MIP) in a straight-forward manner. First, note that if $b_i = 0$ for any $i \in \{1, \dots, m\}$, then the corresponding inequality is redundant since then the inequality is implied by the non-negativity constraints on the x variables. Thus, we assume henceforth that $b_i > 0$ for all $i \in \{1, \dots, m\}$ and so they can be scaled to 1. Then an MIP formulation of CKVLP is

$$\begin{aligned}
\min \quad & c^\top x \\
\text{s.t.} \quad & a_i^\top x + z_i \geq 1 \quad i = 1, \dots, m, \\
& \sum_{i=1}^m z_i \leq k \\
& x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad i = 1, \dots, m,
\end{aligned} \tag{10}$$

where we have introduced the binary variables z_i taking the value 1 if the i -th constraint is violated. For large scale CKVLPs, the above MIP formulation performs very poorly. The goal of this chapter is to study a number of enhancement schemes

to improve the computational performance of MIP-based approaches for solving CK-VLPs.

We begin by studying the theoretical complexity of CKVLPs and illustrating the difficulty of solving realistic instances directly by the CPLEX MIP solver (Section 2.2) as well. Next, in order to improve the performance of standard solvers on the MIP model (10) of CKVLPs, we introduce and analyze a coefficient strengthening scheme (Section 2.3), adapt and analyze an existing cutting plane technique (Section 2.4), and present a branching technique (Section 2.5). Through computational experiments on the probabilistic portfolio optimization problem (9) and an optimal vaccination allocation problem, we empirically verify that these techniques are extremely effective in improving solution times (Section 2.6). In particular, we observe that the proposed schemes can cut down solution times from as much as six days to under four hours in some instances.

We close this section by pointing out that all three enhancement schemes studied here are applicable when there are additional side constraints in the MIP (10). This follows since these schemes attempt to tighten the LP relaxation of (10), which is a valid relaxation even when additional side constraints are present.

2.2 Difficulty of Solving CKVLP

2.2.1 Computational Complexity

General KVLP has been shown to be \mathcal{NP} -complete [9]. However, to the best of our knowledge, the complexity of CKVLP, a sub-class of KVLP, has not been addressed. In a recent paper [116], Tunçel et al. showed that a packing version KVLP is weakly \mathcal{NP} -hard (the linear inequalities in KVLP are packing inequalities) by reduction from the partition problem. This result can be modified to show the \mathcal{NP} -hardness of CKVLP. In this chapter we provide a direct proof that CKVLP is strongly \mathcal{NP} -hard.

By complementing the binary variables z in (10), we have the following equivalent

formulation of CKVLP:

$$\begin{aligned}
& \min && c^\top x \\
& \text{s.t.} && Ax \geq z \\
& && e^\top z \geq p \\
& && x \in \mathbb{R}_+^n \\
& && z \in \{0, 1\}^m,
\end{aligned} \tag{11}$$

where $A = [a_1^\top, \dots, a_m^\top] \in \mathbb{Q}_+^{m \times n}$, $c \in \mathbb{Q}_+^n$, $p = m - k$, e is the column vector with each entry equal to 1, and \mathbb{Q} is the set of rationals.

To prove that CKVLP (11) is \mathcal{NP} -hard, we first verify that the following intermediate decision problem is \mathcal{NP} -complete.

Intermediate CKVLP Feasibility Problem: Given $\eta \in \mathbb{Q}$, $A \in \mathbb{Q}_+^{m \times n}$ and $c \in \mathbb{Q}_+^n$, does there exist a solution $(x, z) \in \mathbb{R}_+^n \times \{0, 1\}^m$ to the following system?

$$\begin{aligned}
c^\top x - e^\top z &\leq \eta \\
Ax &\geq z.
\end{aligned} \tag{12}$$

Lemma 1. *The Intermediate CKVLP Feasibility Problem (12) is strongly \mathcal{NP} -complete.*

Proof. Since (12) is a decision version of a mixed integer linear program, it is in \mathcal{NP} . In order to show that determining the feasibility of (12) is strongly \mathcal{NP} -complete, we polynomially reduce an arbitrary instance of the strongly \mathcal{NP} -complete vertex cover problem [57] to an instance of (12). An instance of the vertex cover problem is defined as follows:

Vertex Cover: Given a graph $G = (V, E)$ and $q \in \mathbb{N}$, does there exist $S \subseteq V$ such that (i) $|S| \leq q$ and (ii) S is a vertex cover, that is for all $(i, j) \in E$, either $i \in S$ or $j \in S$?

Given an instance of the vertex cover problem, we construct an instance of (12) by setting $m := |V| + |E|$, $n := |V|$, $\eta := q - |E|$, $c := 2e$, $A := \begin{bmatrix} H \\ I \end{bmatrix}$, where H is the node-arc incidence matrix of G and I is a $|V| \times |V|$ identity matrix. The resulting instance of (12) is then:

$$2 \sum_{j \in V} x_j - \sum_{j \in V} z_j - \sum_{(i,j) \in E} y_{ij} \leq q - |E| \quad (13)$$

$$x_i + x_j \geq y_{ij} \quad \forall (i, j) \in E \quad (14)$$

$$x_i \geq z_i \quad \forall i \in V \quad (15)$$

$$x \in \mathbb{R}_+^{|V|} \quad (16)$$

$$z \in \{0, 1\}^{|V|} \quad (17)$$

$$y \in \{0, 1\}^{|E|}. \quad (18)$$

Note that the size of (13)-(18) is polynomial in the encoding length of G and q . We complete the proof by showing that a vertex cover instance has an answer yes if and only if the associated system (13)-(18) has a solution.

(\Rightarrow) Let S be a vertex cover for G such that $|S| \leq q$. Then, consider a solution $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}_+^{|V|} \times \{0, 1\}^{|E|} \times \{0, 1\}^{|V|}$ defined as:

$$\begin{aligned} \tilde{x}_j = \tilde{z}_j &= \begin{cases} 1 & \forall j \in S \\ 0 & \forall j \in V \setminus S, \end{cases} \\ \tilde{y}_{i,j} &= 1 \quad \forall (i, j) \in E. \end{aligned}$$

The solution $(\tilde{x}, \tilde{y}, \tilde{z})$ satisfies (15)-(18) by construction, and since S is a vertex cover it also satisfies (14). Finally, $2 \sum_{j \in V} \tilde{x}_j - \sum_{j \in V} \tilde{z}_j - \sum_{(i,j) \in E} \tilde{y}_{ij} = |S| - |E| \leq q - |E|$. Thus the system (13)-(18) has a solution.

(\Leftarrow) Now assume that the system (13)-(18) has a solution. Note that an arbitrary feasible solution to (13)-(18) may have fractional x components that cannot be directly

converted to a vertex cover for G . We show that there exists a feasible solution of (13)-(18) with integral values of x and $y = e$ whenever (13)-(18) is feasible. Towards this end, we first present some properties of feasible solutions to (13)-(18). Given $(x, y, z) \in \mathbb{R}_+^{|V|} \times \{0, 1\}^{|E|} \times \{0, 1\}^{|V|}$, which satisfies (14)-(18), let

$$f(x, y, z) := 2 \sum_{j \in V} x_j - \sum_{j \in V} z_j - \sum_{(i,j) \in E} y_{ij},$$

i.e., if (x, y, z) is feasible for (13)-(18), then $f(x, y, z) \leq q - |E|$.

Claim a. Given (x^1, y^1, z^1) satisfying (14)-(18), there exists (x^2, y^2, z^2) satisfying (14)-(18) such that $y^2 = e$ and $f(x^2, y^2, z^2) \leq f(x^1, y^1, z^1)$.

Proof of Claim a. Suppose there exists $(\tilde{i}, \tilde{j}) \in E$ such that $y_{\tilde{i}\tilde{j}}^1 = 0$. Construct (x^3, y^3, z^3) as follows:

$$\begin{aligned} x_j^3 &= \begin{cases} 1 & j = \tilde{i} \\ x_j^1 & j \in V \setminus \{\tilde{i}\} \end{cases}, \\ z_j^3 &= \begin{cases} 1 & j = \tilde{i} \\ z_j^1 & j \in V \setminus \{\tilde{i}\} \end{cases}, \\ y_{ij}^3 &= \begin{cases} 1 & (i, j) = (\tilde{i}, \tilde{j}) \\ y_{ij}^1 & (i, j) \in E \setminus \{(\tilde{i}, \tilde{j})\}. \end{cases} \end{aligned}$$

It is easy to see that (x^3, y^3, z^3) satisfies (14)-(18). We observe that $f(x^1, y^1, z^1) - f(x^3, y^3, z^3) = (2x_{\tilde{i}}^1 - z_{\tilde{i}}^1 - y_{\tilde{i}, \tilde{j}}^1) - (2 \times 1 - 1 - 1) = x_{\tilde{i}}^1 + (x_{\tilde{i}}^1 - z_{\tilde{i}}^1) \geq 0$, where the last inequality holds due to the fact that (x^1, y^1, z^1) satisfies (15). By repeating the above construction at most $|E|$ times we arrive at a solution (x^2, y^2, z^2) satisfying the claim. \diamond

We now restrict our attention to feasible solutions of (13)-(18) with the vector y fixed to e . Next, we show that a feasible solution with integral x components exists.

Claim b. Given (x^1, e, z^1) satisfying (14)-(18), there exists a solution (x^2, e, z^2) satisfying (14)-(18) such that $x^2 \in \{0, 1\}^{|V|}$ and $f(x^2, e, z^2) \leq f(x^1, e, z^1)$.

Proof of Claim b. If $x^1 \in \{0, 1\}^{|V|}$, then there is nothing to verify. If there exists j such that $x_j^1 > 1$, then we can set $x_j^1 = 1$. The resulting solution still satisfies (14)-(18), and the value of the function f reduces. Therefore, the non-trivial case is when there exists j such that $x_j^1 \in (0, 1)$. We construct a solution (x^2, e, z^2) as follows:

$$\begin{aligned} x_j^2 &= \begin{cases} x_j^1 & j : x_j^1 \in \{0, 1\} \\ 1 & j : 1/2 \leq x_j^1 < 1 \\ 0 & j : 0 < x_j^1 < 1/2 \end{cases} \\ z_j^2 &= \begin{cases} z_j^1 & j : x_j^1 \in \{0, 1\} \text{ or } 0 < x_j^1 < 1/2 \\ 1 & j : 1/2 \leq x_j^1 < 1 \end{cases} . \end{aligned}$$

It is easy to see that (x^2, e, z^2) constructed as above satisfies (14)-(18) since there is no $(i, j) \in E$ such that $x_i^1 < 1/2$ and $x_j^1 < 1/2$. Also each component of x^2 is integral. Now we verify that (x^2, e, z^2) satisfies (13) too. $f(x^1, e, z^1) - f(x^2, e, z^2) = 2 \sum_{j: 0 < x_j^1 < 1/2} x_j^1 + 2 \sum_{j: 1/2 \leq x_j^1 < 1} (x_j^1 - 1) + |\{j : 1/2 \leq x_j^1 < 1\}| \geq 0$. Therefore, (x^2, e, z^2) is a solution we desire.

From the claims a and b, it is clear that there exists a feasible solution of the form (x, y, z) with (i) $y = e$ and (ii) $x \in \{0, 1\}^{|V|}$. If $x_j = 1$ and $z_j = 0$ for some j , then we can set $z_j = 1$, and the resulting solution is still feasible for (13)-(18). Therefore, we may assume that the feasible solution also satisfies $x_j = z_j$ for all $j \in V$. We select any such feasible solution and let $S = \{j : x_j = 1\}$. Clearly, S is a vertex cover for G since $y = e$. Notice that $f(x, y, z) = 2|S| - |S| - |E| \leq q - |E|$ or equivalently $|S| \leq q$. \square

Theorem 1. *CKVLP is strongly \mathcal{NP} -hard.*

Proof. To verify that (11) is \mathcal{NP} -hard, we show that if there exists a polynomial time algorithm for solving (11), then there is a polynomial time algorithm for deciding the feasibility of (12). This completes the proof, since by Lemma 1, we have that deciding the feasibility of (12) is \mathcal{NP} -complete.

Let $v(p)$ denote the optimal value of (11) as a function of $p \in \{0, \dots, m\}$. Consider the following algorithm for deciding the feasibility of (12):

1. Given $A \in \mathbb{Q}_+^{m \times n}$, $c \in \mathbb{Q}^n$, and $\eta \in \mathbb{Q}$, compute $v(p)$ for all $p \in \{0, \dots, m\}$, using the polynomial-time algorithm for solving (11).
2. Compute $\eta^* := \min_{0 \leq p \leq m} \{v(p) - p\}$. If $\eta^* \leq \eta$, return “yes,” (i.e. (12) is feasible); otherwise return “no.”

Notice that the above algorithm is a polynomial time algorithm in the size of the encoding of (12). It remains to verify the validity of the above algorithm.

Suppose $\eta^* \leq \eta$ and $p^* \in \operatorname{argmin}\{v(p) - p\}$. Consider an optimal solution (x^*, z^*) to (11) corresponding to $p = p^*$. Since $\eta \geq \eta^* = v(p^*) - p^* \geq v(p^*) - e^\top z^* = c^\top x^* - e^\top z^*$, the instance of (12) is feasible.

Suppose $\eta^* > \eta$. Assume to the contrary that the instance of (12) is feasible and let (x^*, z^*) be a feasible point. Let $p^* = \sum_{j=1}^m z_j^*$. Then, observe that (x^*, z^*) is feasible to (11) corresponding to $p = p^*$. Thus, $\eta^* \leq v(p^*) - p^* \leq c^\top x^* - p^* \leq \eta$, a contradiction. \square

2.2.2 Performance of a standard MIP solver on CKVLP instances

Given the significant advancements made in MIP solution technology, many instances of \mathcal{NP} -hard problems are not necessarily difficult to solve in practice. To assess the practical computational difficulty of CKVLP, we next report on the performance of CPLEX, a state-of-the-art MIP solver, on randomly generated instances of the MIP (10).

We consider instances with $n = 20$, $m = 200$ and $k \in \{15, 20\}$. The data is generated as follows:

1. “Dense Data”: Each left-hand-side coefficient a_{ij} is generated uniformly between 0.8 and 1.5, and then the coefficients are divided by 1.1. The cost vector is e .
2. “Sparse Data”: This uses the same input data as in “Dense Data”, except that half of the left-hand-side coefficients are randomly set to zero.
3. “Random Objective”: These instances have the same constraint coefficients as in “Dense Data”, but with random integer cost coefficients between 1 and 100.

For each of the six combinations of two values of k and three data classes, we considered 10 instances giving a total of 60 instances. The computations are run on Intel Xeon 2.27 GHz dual core Linux server installed with 4 Gb RAM. The model is implemented with the callable libraries and solved by the MIP solver in CPLEX 12.1 with default settings.

The average results over ten instances in each size-data combination are presented in Table 1. The ‘Gap Closed’ column in the table reports the root node LP relaxation gap closed by CPLEX cuts. That is, the value $\left(\frac{z^{LP+Cuts} - z^{LP}}{z^* - z^{LP}} \right) \times 100$, where $z^{LP+Cuts}$, z^{LP} , and z^* are the objective function values of the LP relaxation with CPLEX cuts at the root node, of just the LP relaxation, and of the MIP, respectively. The ‘Nodes’ and the ‘Time’ columns report the number of branch-and-bound tree nodes generated and the time in seconds needed to solve the instances to optimality, respectively.

Table 1: Performance of CPLEX on CKVLP

	Dense Data			Sparse Data			Random Objective		
k	Gap Closed	Nodes	Time	Gap Closed	Nodes	Time	Gap Closed	Nodes	Time
15	2%	3,537,864	2,454	7%	158,039	83	17%	1,777	1
20	2%	43,296,679	25,948	6%	1,769,574	917	21%	6,227	2

Following are some observations based on the above computations.

1. The effect of k : Setting k to a larger value results in a substantial increase in time and memory consumption (measured in the number of nodes in the branch-and-bound tree), as seen by a ten-fold increase for the first two types of instances. This phenomenon can perhaps be explained by the combinatorial nature of CKVLP, which is to choose the linear program with the best objective value among $\binom{m}{k}$ linear programs. When k increases to $\lfloor \frac{m}{2} \rfloor$, the number of possible linear programs increases rapidly.
2. The effect of sparsity: The coefficient matrix density, measured by the number of non-zeros, can make instances significantly harder to solve, as seen by a 20-fold increase in nodes and 30-fold increase in time when the density increases from 50% to 100%. The dense coefficients not only make the LP relaxation hard to solve, but also make it hard for CPLEX to find effective cuts, e.g., CPLEX default cuts close only 2% of the LP relaxation gap in the “Dense Data” instances, whereas 6-7% of the gap is closed in the “Sparse Data” instances.
3. The effect of objective function: The objective function coefficients play a crucial role in determining the computational difficulty, as demonstrated by the contrast between “Dense Data” and “Random Objective”. The instances with random objective coefficients can be solved in seconds; however, the instances with the same constraints but uniform objective coefficients in “Dense Data” take hours to solve. When the cost coefficients and the constraint coefficients are set up in a way so that the objective values of linear programs formed by different choices of linear constraints are close, the branch-and-bound procedure generates a great number of nodes, of which the LPs are similar in terms of bounds, and the MIP solver spends an enormous amount of time on proving optimality.

In the rest of the chapter, we focus on variants of the most difficult class of the

above instances, that is, instances that are very similar in type to “Dense Data,” and attempt to tighten the root node lower bound and reduce the size of the search tree.

2.3 Iterative Coefficient Strengthening

In this section, we propose and analyze a scheme for strengthening the coefficients of the binary variables in the MIP formulation (10) of CKVLP. Let X denote the set of feasible x solutions of (10), i.e.

$$X := \{x \in \mathbb{R}_+^n : \exists z \in \{0, 1\}^m \text{ s.t. } a_i^\top x + z_i \geq 1 \ \forall i = 1, \dots, m \text{ and } \sum_{i=1}^m z_i \leq k\}. \quad (19)$$

Definition 1. A vector $\ell \in \mathbb{R}^m$ is called a valid bound vector if $\ell_i \leq \min\{a_i^\top x : x \in X\}$ for all $i = 1, \dots, m$.

Given a valid bound vector ℓ , let

$$X(\ell) := \{x \in \mathbb{R}_+^n : \exists z \in [0, 1]^m \text{ s.t. } a_i^\top x + (1 - \ell_i)z_i \geq 1 \ \forall i = 1, \dots, m \text{ and } \sum_{i=1}^m z_i \leq k\}.$$

Proposition 2. (i) If ℓ is a valid bound vector then $X(\ell) \supseteq X$. (ii) The bound vector $\ell = 0$ is valid. (iii) For valid bounds ℓ^1 and ℓ^2 , if $\ell^2 \geq \ell^1$ then $X(\ell^1) \supseteq X(\ell^2)$.

Proof. (i) If $x \in X$ then there exists $z \in \{0, 1\}^m$ such that $a_i^\top x \geq \max\{1 - z_i, \ell_i\}$ for all $i = 1, \dots, m$ and $\sum_{i=1}^m z_i \leq k$. Since $\max\{1 - z_i, \ell_i\} = 1 - (1 - \ell_i)z_i$ when $z_i \in \{0, 1\}$, it follows that $a_i^\top x + (1 - \ell_i)z_i \geq 1$ and $x \in X(\ell)$. (ii) Since $a^\top x \geq 0$ for all $x \in \mathbb{R}_+^n$, we obtain that $\ell = 0$ is a valid bound vector. (iii) If $x \in X(\ell^2)$ then there exists $z \in [0, 1]^m$ such that $a_i^\top x \geq 1 - (1 - \ell_i^2)z_i$ for all $i = 1, \dots, m$ and $\sum_{i=1}^m z_i \leq k$. Since $z_i \geq 0$ this implies that $a_i^\top x \geq 1 - (1 - \ell_i^1)z_i$, hence $x \in X(\ell^1)$. \square

Note that $X(0)$ is the projection, on to the x variables, of the LP relaxation of the MIP formulation (10). Proposition 2 suggests that we can strengthen this LP relaxation by iteratively tightening the bound vector ℓ and hence the coefficients of the binary variables in (10), starting from $\ell = 0$. Algorithm 1 describes such

Algorithm 1 Iterative Coefficient Strengthening

Input : A threshold parameter $\varepsilon > 0$ and the data (m, n, k, a_{ij}) describing X

Output : A valid bound vector $\hat{\ell} \in \mathbb{R}_+^m$

```
 $\Delta \leftarrow 2\varepsilon, t \leftarrow 1, \ell^t \leftarrow 0$ 
while  $\Delta > \varepsilon$  do
  for  $i = 1, \dots, m$  do
     $\ell_i^{t+1} = \min\{a_i^\top x : x \in X(\ell^t)\}$ 
  end for
   $\Delta \leftarrow \|\ell^{t+1} - \ell^t\|_\infty$ 
   $t \leftarrow t + 1$ 
end while
 $\hat{\ell} \leftarrow \ell^t$ 
```

a coefficient strengthening procedure. Note that this procedure requires solving m feasible linear programs with bounded objectives in each iteration.

Proposition 3. *Let $\{\ell^t\}$ be the sequence of bound vectors produced in Algorithm 1. We have (i) $\ell^{t+1} \geq \ell^t$ and (ii) ℓ^t is a valid bound vector for all t . Accordingly, Algorithm 1 terminates finitely returning a valid bound vector $\hat{\ell}$.*

Proof. We proceed by induction on t . For the base case $t = 1$ we have $\ell^1 = 0$, then (ii) holds from part (ii) of Proposition 2. Moreover $\ell_i^2 = \min\{a_i^\top x : x \in X(0)\} \geq 0$ for all i , hence (i) holds. Suppose now that (i) and (ii) hold for some $t > 1$. By definition $\ell_i^{t+1} = \min\{a_i^\top x : x \in X(\ell^t)\}$ for all $i = 1, \dots, m$. Thus, for each $i = 1, \dots, m$, $\ell_i^{t+1} \leq a_i^\top x$ for all $x \in X(\ell^t)$ and hence for all $x \in X$ since $X \subseteq X(\ell^t)$ from the validity of ℓ^t . Thus ℓ^{t+1} is a valid bound vector and (ii) holds for all t . By our induction hypothesis $\ell^{t+1} \geq \ell^t$ thus $X(\ell^{t+1}) \subseteq X(\ell^t)$ by part (iii) of Proposition 2. Thus $\ell_i^{t+2} = \min\{a_i^\top x : x \in X(\ell^{t+1})\} \geq \min\{a_i^\top x : x \in X(\ell^t)\} = \ell_i^{t+1}$ for all $i = 1, \dots, m$, and so (i) holds for all t . Finally note that, for any t , $X(\ell^t) \supseteq X$ from part (i) of Proposition 2, thus $\ell_i^t = \min\{a_i^\top x : x \in X(\ell^t)\} \leq \min\{a_i^\top x : x \in X\} =: \bar{\ell}_i^*$, where $\bar{\ell}_i^*$ is a well defined finite value, for all $i = 1, \dots, m$. Thus, for each $i = 1, \dots, m$, $\{\ell_i^t\}$ is a bounded nondecreasing sequence, and hence convergent. It follows that for any $\varepsilon > 0$ there exists a sufficiently large value of t such that $\|\ell^{t+1} - \ell^t\|_\infty \leq \varepsilon$ ensuring

finite termination of the algorithm. □

Next we analyze the strength of the LP relaxation of (10) using tightened coefficients derived using Algorithm 1. Given a cost vector c , let

$$v^* = \min\{c^\top x : x \in X\} \text{ and } v^L(\ell) = \min\{c^\top x : x \in X(\ell)\}, \quad (20)$$

be the optimal value of the MIP (10) and the optimal value of the LP relaxation corresponding to bound vector ℓ , respectively. Note that these values are finite as long as $c \geq 0$. Recall that $v^L(0)$ is the natural LP relaxation bound for (10), and the coefficient tightening scheme in Algorithm 1 is aimed to improve this bound. In the following we analyze this improvement as a function of the instance data. Let

$$\rho_i = \min_{j=1,\dots,n} \left\{ \frac{a_{ij}}{(1/m) \sum_{i'=1}^m a_{i'j}} \right\} \quad (21)$$

and

$$\rho = \min_{i=1,\dots,m} \rho_i. \quad (22)$$

Note that ρ is a measure of the variability of the constraint coefficient data and $\rho \in (0, 1]$. Let $\{\ell^t\}$ be the sequence of bound vectors produced by the scheme in Algorithm 1 with a threshold of $\varepsilon = 0$. From Proposition 3 we know that this sequence is convergent. Let

$$\ell^* = \lim_{t \rightarrow \infty} \ell^t. \quad (23)$$

Lemma 2. *Assuming $a_{ij} > 0$ for all $i = 1, \dots, m$ and $j = 1, \dots, n$,*

$$\ell_i^* \geq \frac{m-k}{m-\rho k} \rho_i \quad \forall i = 1, \dots, m,$$

where ρ_i, ρ and ℓ^* are as defined in (21) and (23), respectively.

Proof. Let $\{u^t\}$ be a sequence of scalars defined by the following recursion:

$$u^1 = 0 \text{ and } u^{t+1} = (1 - (1 - u^t)k/m) \quad \forall t \geq 1. \quad (24)$$

First, we claim that

$$\ell^t \geq u^t e \geq 0 \quad \forall t \geq 1, \quad (25)$$

We prove this claim by induction on t . Note that (25) holds for $t = 1$ since $\ell_i^1 = u^1 = 0$ for all $i = 1, \dots, m$. Suppose now that (25) holds for some $t > 1$. Since $u^t \geq 0$ and $0 < k/m \leq 1$ we have that $(1 - (1 - u^t)k/m) = (1 - k/m) + u^t k/m \geq 0$, and hence $u^{t+1} \geq 0$. Let $\mu_j = \sum_{i=1}^m a_{ij}/m$ for $j = 1, \dots, n$ and μ be the corresponding n -dimensional vector. For any $i = 1, \dots, m$,

$$\ell_i^{t+1} = \min\{a_i^\top x : x \in X(\ell^t)\} \quad (26)$$

$$\geq \min\{a_i^\top x : x \in X(u^t e)\} \quad (27)$$

$$= \min\{a_i^\top x : a_{i'}^\top x + (1 - u^t)z_{i'} \geq 1 \quad \forall i' = 1, \dots, m, \sum_{i'=1}^m z_{i'} \leq k, \\ x \in \mathbb{R}_+^n, z \in [0, 1]^m\} \quad (28)$$

$$\geq \min\{a_i^\top x : \mu^\top x \geq 1 - (1 - u^t)k/m, x \in \mathbb{R}_+^n\} \quad (29)$$

$$= (1 - (1 - u^t)k/m) \left(\min_{j=1, \dots, n} \{a_{ij}/\mu_j\} \right) = (1 - (1 - u^t)k/m) \rho_i \quad (30)$$

$$\geq (1 - (1 - u^t)k/m) \rho \quad (31)$$

$$= u^{t+1}, \quad (32)$$

where (27) follows from the induction hypothesis $\ell^t \geq u^t e$ since $X(\ell^t) \subseteq X(u^t e)$ by Proposition 2(iii); (28) follows from the definition of $X(u^t)$; (29) follows by aggregating the m rows of the linear program defined in (28) and eliminating the z variables; since $(1 - (1 - u^t)k/m) \geq 0$, (30) follows from the optimal solution of the single constrained linear program defined in (29); (31) follows from the definition of ρ ; and (32) follows from the definition of u^{t+1} . Thus (25) holds.

Next we claim that, for all $i = 1, \dots, m$, $\{u^t\}$ is convergent and

$$\lim_{t \rightarrow \infty} u^t = \frac{m - k}{m - \rho k} \rho. \quad (33)$$

Consider any $i \in \{1, \dots, m\}$. We first verify that $u^t \leq \frac{m-k}{m-\rho k} \rho$ for all t . We proceed by induction on t . By definition $u^1 = 0 \leq \frac{m-k}{m-\rho k} \rho$. By induction hypothesis, we have that

$u^t \leq \frac{m-k}{m-\rho k} \rho$. Now $u^{t+1} = \rho - \rho \frac{k}{m} + \rho \frac{k}{m} u^t \leq \rho - \rho \frac{k}{m} + \rho \frac{k}{m} \left(\frac{m-k}{m-\rho k} \rho \right) = \frac{m-k}{m-\rho k} \rho$. Now we verify that the sequence $\{u^t\}$ is non-decreasing. Observe that $u^t - u^{t+1} = u^t - \left(\rho - \rho \frac{k}{m} + \rho \frac{k}{m} u^t \right) = u^t \left(1 - \rho \frac{k}{m} \right) - \rho + \rho \frac{k}{m} \leq \left(\frac{m-k}{m-\rho k} \rho \right) \left(1 - \rho \frac{k}{m} \right) - \rho + \rho \frac{k}{m} = 0$. Finally suppose by contradiction that the sequence $\{u^t\}$ converges to a value $\frac{m-k}{m-\rho k} \rho - \delta$, where $\delta > 0$. Therefore, there exists a t such that $\frac{m-k}{m-\rho k} \rho - \delta > u^t > \frac{m-k}{m-\rho k} \rho - \delta - \epsilon$, in which $\epsilon = \delta \left(1 - \rho \frac{k}{m} \right) > 0$. Since $\rho \frac{k}{m} < 1$, we have $\left(1 - \rho \frac{k}{m} \right) < 1$. Hence, we obtain $u^t - u^{t+1} < \left(\frac{m-k}{m-\rho k} \rho - \delta \right) \left(1 - \rho \frac{k}{m} \right) - \rho + \rho \frac{k}{m} < -(\delta) \left(1 - \rho \frac{k}{m} \right) = -\epsilon$. Thus, $u^{t+1} > u^t + \epsilon > \frac{m-k}{m-\rho k} \rho - \delta$ which is a contradiction. Thus (33) holds.

It then follows from (25) and (33) that

$$\ell_i^* \geq \frac{m-k}{m-\rho k} \rho \quad \forall i = 1, \dots, m.$$

Since u^t converges to $\rho \frac{m-k}{m-\rho k}$ when $t \rightarrow \infty$, it follows from (30) that

$$\ell_i^* \geq \frac{m-k}{m-\rho k} \rho_i \quad \forall i = 1, \dots, m.$$

□

Theorem 4. Assuming $c_j > 0$ and $a_{ij} > 0$ for all $i = 1, \dots, m$ and $j = 1, \dots, m$,

$$\frac{v^* - v^L(\ell^*)}{v^*} \leq \frac{m(1-\rho)}{m-\rho k}. \quad (34)$$

Proof. Note that

$$\begin{aligned} v^L(\ell^*) &= \min\{c^\top x : a_i^\top x + (1 - \ell_i^*)z_i \geq 1 \quad \forall i = 1, \dots, m, \sum_{i=1}^m z_i \leq k, \\ &\quad x \in \mathbb{R}_+^n, z \in [0, 1]^m\} \end{aligned} \quad (35)$$

$$\begin{aligned} &\geq \min\{c^\top x : a_i^\top x + \left(1 - \frac{m-k}{m-\rho k} \rho\right) z_i \geq 1 \quad \forall i = 1, \dots, m, \sum_{i=1}^m z_i \leq k, \\ &\quad x \in \mathbb{R}_+^n, z \in [0, 1]^m\} \end{aligned} \quad (36)$$

$$\geq \min\{c^\top x : \mu^\top x + \left(1 - \frac{m-k}{m-\rho k} \rho\right) \frac{k}{m} \geq 1 \quad \forall i = 1, \dots, m, x \in \mathbb{R}_+^n\} \quad (37)$$

$$= \frac{c_{\hat{j}}}{\mu_{\hat{j}}} \frac{m-k}{m-\rho k} \quad (38)$$

where

$$\hat{j} \in \operatorname{argmin} \left\{ \frac{c_j}{\mu_j} : j = 1, \dots, n \right\}. \quad (39)$$

In the above, (36) follows from Lemma 2; (37) follows from aggregating the rows of the LP defined in (36) and eliminating the z variables; and (38) follows from solving the single constrained LP defined in (37).

Note that

$$v^* = \min \left\{ c^\top x : a_i^\top x + z_i \geq 1 \ \forall i \in \{1, \dots, m\}, \sum_{i=1}^m z_i \leq k, x \in \mathbb{R}_+^n, z \in \{0, 1\}^m \right\}.$$

Next we obtain an upper bound on v^* . For \hat{j} defined in (39):

1. Sort a_{ij} 's from smallest to largest.
2. Let $a_{i\hat{j}}$ be the $(k+1)^{\text{st}}$ smallest number.
3. Let $v^H = \frac{c_{\hat{j}}}{a_{i\hat{j}}}$. This corresponds to the objective function value of the feasible solution $x_j = 0$ for $j \neq \hat{j}$ and $x_{\hat{j}} = \frac{1}{a_{i\hat{j}}}$. Thus $v^* \leq v^H$.

Now observe that

$$\frac{c_{\hat{j}}}{\mu_{\hat{j}}} \frac{m-k}{m-\rho k} \leq v^L \leq v^* \leq \frac{c_{\hat{j}}}{a_{i\hat{j}}} = v^H. \quad (40)$$

Therefore using the definition of ρ we obtain that,

$$\frac{v^* - v^L}{v^*} \leq \frac{v^H - v^L}{v^H} \leq \frac{m(1-\rho)}{m-\rho k}. \quad (41)$$

□

If there is $a_{ij} = 0$ for some i and j , then the minimization problem in (27) yields zero, which is the original trivial lower bound. In this case $\rho = 0$ and the upper bound result in Theorem 4 is trivial. However, note that the CKVLP is motivated mainly by the sample average approximations of probabilistic covering problems where the constraint coefficients are nonnegative random variables. In case of continuous distributions the sampled coefficients will be positive with probability one. Thus, the assumption of $a_{ij} > 0$ for all i and j is not too restrictive in this context.

2.4 Mixing Set Inequalities

In this section, we study valid inequalities derived from a 0-1 version of mixing set relaxation of CKVLP. A *0-1 mixing set* is defined as follows:

$$P = \{(y, z) \in \mathbb{R}_+ \times \{0, 1\}^m : y + h_i z_i \geq h_i, i = 1, \dots, m\}, \quad (42)$$

where $h_1 \geq h_2 \geq \dots \geq h_m$. The more general version of mixing set, in which z_i s are general integers, was introduced by Günlük and Pochet [63], and its variants in different contexts have also been studied in [37, 92, 38, 119, 62, 76]. The following inequalities, known as *mixing (set) inequalities*, are valid for P :

$$y + \sum_{j=1}^l (h_{t_j} - h_{t_{j+1}}) z_{t_j} \geq h_{t_1} \quad \forall T = \{t_1, \dots, t_l\} \subseteq \{1, \dots, m\}, \quad (43)$$

where $h_{t_1} > h_{t_2} > \dots > h_{t_l}$ and $h_{t_{l+1}} := 0$. Furthermore, these inequalities can be separated in polynomial time, are facet-defining for P when $t_1 = 1$, and are sufficient to describe the convex hull of P [13, 63].

Recently, the mixing set inequalities have been applied to solve the MIP formulation of chance-constrained problems, in which the formulation has a k -violation-type substructure, i.e., a feasible solution must satisfy the constraints corresponding to at least k out of m scenarios [82, 83, 84]. CKVLPs can be viewed as a special case of this substructure in which each scenario consists of only one covering linear constraint. We next describe and analyze the mixing set inequalities for CKVLPs.

Let the set of (x, z) -solutions to the MIP (10) be denoted by X_{MIP} , and recall from (19) that the set of x -solutions to (10) is denoted by X . Note that X is the projection of X_{MIP} into x -space, i.e., $X = \text{Proj}_x(X_{\text{MIP}})$. Following [83], we can obtain a mixing set relaxation of X_{MIP} as follows. Given a vector $\alpha \in \mathbb{R}_+^n$, calculate $\beta_i^\alpha, i \in \{1, \dots, m\}$ as below:

$$\beta_i^\alpha := \min \{\alpha^\top x : a_i^\top x \geq 1, x \in \mathbb{R}_+^n\}, \quad (44)$$

where a_i is the coefficient vector for the i -th constraint in the MIP (10). Assume

without loss of generality that $\beta_1^\alpha \geq \beta_2^\alpha \geq \dots \geq \beta_m^\alpha$, and consider the following set

$$Y(\alpha) := \{(x, z) \in \mathbb{R}_+^n \times \{0, 1\}^m : \alpha^\top x + (\beta_i^\alpha - \beta_{k+1}^\alpha)z_i \geq \beta_i^\alpha, \quad i = 1, \dots, k\}. \quad (45)$$

Proposition 5. *For any $\alpha \in \mathbb{R}_+^n$, $X_{\text{MIP}} \subseteq Y(\alpha)$ and $X \subseteq \text{Proj}_x(Y(\alpha))$*

Proof. Let $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$. Then the non-negativity constraints and integrality constraints in $Y(\alpha)$ are satisfied by (\bar{x}, \bar{z}) . Without loss of generality, we may assume that the indexes $1, \dots, k$ in $Y(\alpha)$ are the first k indexes in X_{MIP} . It remains to verify that (\bar{x}, \bar{z}) satisfies the constraints $\alpha^\top \bar{x} + (\beta_i^\alpha - \beta_{k+1}^\alpha)\bar{z}_i \geq \beta_i^\alpha$ for all $i = 1, \dots, k$.

- (i) For i such that $\bar{z}_i=1$: We require to verify that $\alpha^\top \bar{x} \geq \beta_{k+1}^\alpha$. Since $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$, there exists some $u \in \{1, \dots, k+1\}$ such that $a_u^\top \bar{x} \geq 1$. Moreover as $\beta_u^\alpha = \min\{\alpha^\top x : x \in \mathbb{R}_+^n, a_u^\top x \geq 1\}$, we obtain that $\alpha^\top \bar{x} \geq \beta_u^\alpha \geq \beta_{k+1}^\alpha$, where the last inequality is due to the fact that $u \leq k+1$.
- (ii) For i such that $\bar{z}_i=0$: We require to verify that $\alpha^\top \bar{x} \geq \beta_i^\alpha$. Since $(\bar{x}, \bar{z}) \in X_{\text{MIP}}$, we obtain that $a_i^\top \bar{x} \geq 1$. Moreover as $\beta_i^\alpha = \min\{\alpha^\top x : x \in \mathbb{R}_+^n, a_i^\top x \geq 1\}$, we have that $\alpha^\top \bar{x} \geq \beta_i^\alpha$.

Therefore, $(\bar{x}, \bar{z}) \in Y(\alpha)$ and $X_{\text{MIP}} \subseteq Y(\alpha)$. The result $X \subseteq \text{Proj}_x(Y(\alpha))$ follows from the fact that $X = \text{Proj}_x(X_{\text{MIP}})$. \square

The set $Y(\alpha)$ is a valid relaxation of X_{MIP} and it is in the form of a mixing set. This can be noted by considering $y := (\alpha^\top x - \beta_{k+1}^\alpha)$ as a nonnegative continuous variable to obtain the mixing system

$$y + (\beta_i^\alpha - \beta_{k+1}^\alpha)z_i \geq \beta_i^\alpha - \beta_{k+1}^\alpha \quad \forall i = 1, \dots, k.$$

Thus, we have the complete description of $\text{conv}(Y(\alpha))$ using the inequalities (43), which are also valid for X_{MIP} , i.e., $\text{conv}(X_{\text{MIP}}) \subseteq \text{conv}(Y(\alpha))$. Let us call $\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(Y(\alpha))$ the *mixing closure*. Clearly, the mixing closure is a valid relaxation of $\text{conv}(X_{\text{MIP}})$. Let v^{MIX} be the optimal objective value of optimizing over the mixing closure, and v^* be

the optimal objective value of the MIP (10). Then, the best root node gap that can be potentially achieved by the mixing inequality procedure is bounded by $(v^* - v^{\text{MIX}})/v^*$. To study this gap quantitatively, e.g., deriving a bound for $(v^* - v^{\text{MIX}})/v^*$, we analyze the projection of the mixing closure on the x -space, i.e., $\text{Proj}_x(\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(Y(\alpha)))$ in the following subsections.

2.4.1 The basic mixing closure

Note that

$$\begin{aligned} \text{conv}(X) &= \text{Proj}_x(\text{conv}(X_{\text{MIP}})) \subseteq \text{Proj}_x\left(\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(Y(\alpha))\right) \\ &\subseteq \bigcap_{\alpha \in \mathbb{R}_+^n} \text{Proj}_x(\text{conv}(Y(\alpha))) = \bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(\text{Proj}_x(Y(\alpha))). \end{aligned} \quad (46)$$

where the last equality follows from the fact that $\text{Proj}(\text{conv}(\cdot)) = \text{conv}(\text{Proj}(\cdot))$. Thus, minimizing over $\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(\text{Proj}_x(Y(\alpha)))$ yields a lower bound for v^{MIX} .

Proposition 6. $\text{Proj}_x(Y(\alpha)) = \{x \in \mathbb{R}_+^n : \alpha^\top x \geq \beta_{k+1}^\alpha\}$.

Proof. \subseteq : Let $\bar{x} \in \text{Proj}_x(Y(\alpha))$, then there exists $\bar{z} \in \{0, 1\}^k$ such that $\alpha^\top \bar{x} + (\beta_i^\alpha - \beta_{k+1}^\alpha)\bar{z}_i \geq \beta_i^\alpha, i = 1, \dots, k$. Thus $\alpha^\top \bar{x} \geq \beta_i^\alpha(1 - \bar{z}_i) + \beta_{k+1}^\alpha \bar{z}_i \geq \beta_{k+1}^\alpha$ since $\beta_i^\alpha \geq \beta_{k+1}^\alpha$ and $\bar{z}_i \in [0, 1]$.

\supseteq : Let $\bar{x} \in \{x \in \mathbb{R}_+^n : \alpha^\top x \geq \beta_{k+1}^\alpha\}$, set $\bar{z}_i = 1, i = 1, \dots, k$, then $(\bar{x}, \bar{z}) \in Y(\alpha)$ and $\bar{x} \in \text{Proj}_x(Y(\alpha))$. \square

Since $\text{Proj}_x(Y(\alpha))$ is a half space in the non-negative orthant and hence convex, the convex hull operator in $\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(\text{Proj}_x(Y(\alpha)))$ is unnecessary.

Proposition 7.

$$\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(\text{Proj}_x(Y(\alpha))) = \bigcap_{\alpha \in \mathbb{R}_+^n} \text{Proj}_x(Y(\alpha)) = \bigcap_{\alpha \in \mathbb{R}_+^n} \{x \in \mathbb{R}_+^n : \alpha^\top x \geq \beta_{k+1}^\alpha\}.$$

Proposition 7 and (46) indicate that the projection of the mixing closure onto the x -space is contained in the closure constituted by infinitely many half spaces. To study this closure, we give a formal definition as below:

Definition 2 (Basic Mixing Closure). The *Basic Mixing Closure* is defined as

$$\mathcal{M} := \bigcap_{\alpha \in \mathbb{R}_+^n} \{x \in \mathbb{R}_+^n : \alpha^\top x \geq \beta^\alpha\}, \quad (47)$$

where $\beta^\alpha := \beta_{k+1}^\alpha$.

We call $\alpha^\top x \geq \beta^\alpha$ a *basic mixing inequality* corresponding to α . In order to understand the basic mixing closure, we describe another class of inequalities.

Definition 3 (Simple Disjunctive Cuts and Closure).

1. Select a subset S of $k + 1$ constraints. Since at least one of these constraints must be satisfied, we obtain the simple disjunction:

$$(a_{i_1}^\top x \geq 1, x \in \mathbb{R}_+^n) \vee (a_{i_2}^\top x \geq 1, x \in \mathbb{R}_+^n) \vee \cdots \vee (a_{i_{k+1}}^\top x \geq 1, x \in \mathbb{R}_+^n), \quad (48)$$

where $S = \{i_1, \dots, i_{k+1}\}$.

2. Define $a_S \in \mathbb{R}^n$ as

$$(a_S)_j = \max_{i \in S} \{a_{ij}\} \quad \forall j = 1, \dots, n.$$

The convex hull of (48) is

$$(a_S)^\top x \geq 1, x \in \mathbb{R}_+^n,$$

and we call $(a_S)^\top x \geq 1$ a *simple disjunctive cut*.

We define the *simple disjunctive closure* as

$$\mathcal{D} := \bigcap_{S \subseteq \{1, \dots, m\}, |S|=k+1} \{x \in \mathbb{R}_+^n : (a_S)^\top x \geq 1\}. \quad (49)$$

Proposition 8. $\mathcal{D} = \mathcal{M}$

Proof. $\mathcal{D} \subseteq \mathcal{M}$: For any given α , without loss of generality, let $\beta_1 \geq \dots \beta_k \geq \beta_{k+1} \geq \dots \geq \beta_m$. Then $\beta^\alpha = \beta_{k+1}$. Since $\alpha^\top x \geq \beta_i$ is a valid inequality for the set $\{a_i^\top x \geq 1, x \in \mathbb{R}_+^n\} \quad \forall i = 1, \dots, k+1$, $\alpha^\top x \geq \beta^\alpha$ is a valid inequality for the convex hull of the set

$$(a_1^\top x \geq 1, x \in \mathbb{R}_+^n) \vee (a_2^\top x \geq 1, x \in \mathbb{R}_+^n) \vee \dots \vee (a_{k+1}^\top x \geq 1, x \in \mathbb{R}_+^n),$$

or equivalently $\alpha^\top x \geq \beta^\alpha$ is dominated by the inequality $(a_S)^\top x \geq 1$.

$\mathcal{M} \subseteq \mathcal{D}$: Let $S \subseteq \{1, \dots, m\}$ such that $|S| = k+1$. We set $\alpha = \alpha_S$. Then for any $i \in \{1, \dots, m\}$,

$$\begin{aligned} \beta_i &= \min (a_S)^\top x \\ \text{s.t. } & (a_i)^\top x \geq 1, x \in \mathbb{R}_+^n. \end{aligned}$$

Since $a_{ij} \leq (a_S)_j$, we obtain that $\beta_i = \min_{1 \leq j \leq n} \frac{(a_S)_j}{a_{ij}} \geq 1$. Therefore, $\beta^{\alpha_S} \geq 1$. Hence, the basic mixing inequality is

$$(a_S)^\top x \geq \beta^{(\alpha_S)}$$

which dominates the inequality $(a_S)^\top x \geq 1$. □

Because m and k are finite numbers, the number of simple disjunctive cuts is also finite, the following result is immediate:

Corollary 9. \mathcal{M} is polyhedral.

Although the mixing set closure, i.e., $\text{Proj}_x(\bigcap_{\alpha \in \mathbb{R}_+^n} \text{conv}(Y(\alpha)))$, is contained in the basic mixing set closure, which is polyhedral, it remains an open question as to whether the mixing set closure itself is polyhedral.

2.4.2 Bound Quality

Using the equivalence of \mathcal{D} and \mathcal{M} , and the fact that \mathcal{D} has an explicit form and simple structure, we derive a lower bound for v^{MIX} by studying \mathcal{D} . We then provide an upper bound on the worst possible gap achievable by the addition of all possible mixing inequalities, i.e., $(v^* - v^{\text{MIX}})/v^*$.

Proposition 10. *Suppose $c > 0$ and $a_{ij} > 0$ for all i, j . Let $\underline{a} = \min_{ij}\{a_{ij}\}$ and $\bar{a} = \max_{ij}\{a_{ij}\}$. Let v^* be the optimal objective value over X and v^M be the optimal value over the basic mixing closure, then*

$$0 \leq \frac{v^* - v^{\text{MIX}}}{v^*} \leq \frac{v^* - v^M}{v^*} \leq \frac{\bar{a} - \underline{a}}{\bar{a}}.$$

Proof. Let $\underline{c} = \min_j\{c_j\}$. Note that $v^* \leq \min\{c^\top x : a_i^\top x \geq 1 \forall i = 1, \dots, m, x \in \mathbb{R}_+^n\} \leq \min\{c^\top x : (e^\top x) \geq 1/\underline{a}, x \in \mathbb{R}_+^n\} = \underline{c}/\underline{a}$. By the equivalence of \mathcal{D} and \mathcal{M} , we obtain that $v^M = \min\{c^\top x : (a_S)^\top x \geq 1 \forall S \subseteq \{1, \dots, m\}, |S| = k+1, x \in \mathbb{R}_+^n\} \geq \min\{c^\top x : \bar{a}(e^\top x) \geq 1, x \in \mathbb{R}_+^n\} = \underline{c}/\bar{a}$. Thus, $(v^* - v^M)/v^* = 1 - v^M/v^* \leq 1 - (\underline{c}/\bar{a})/(\underline{c}/\underline{a}) = (\bar{a} - \underline{a})/\bar{a}$. \square

The above result implies that the relaxations \mathcal{D} and equivalently \mathcal{M} can be tight when the variation of the constraint coefficients is small. However, the separation of the most violated simple disjunctive cut from \mathcal{D} is \mathcal{NP} -complete. Consider an arbitrary $x^* \in \mathbb{R}_+^n$ that we want to separate. Let $M := \{i \in \{1, \dots, m\} : a_i^\top x^* < 1\}$. Clearly, $|M| > k$, because otherwise, x^* belongs to the feasible region of the k -violation problem X and therefore belongs to \mathcal{D} . When $|M| \geq k+1$, we solve the

following separation problem:

$$\begin{aligned}
\eta = \min \quad & \sum_{j=1}^n \pi_j x_j^* - 1 \\
\text{s.t.} \quad & \pi_j + a_{ij} w_i \geq a_{ij} \quad j = 1, \dots, n; \quad \forall i \in M \\
& \sum_{i \in M} w_i = |M| - (k + 1) \\
& \pi_j \geq 0 \quad \forall j \in \{1, \dots, n\} \\
& w_i \in \{0, 1\} \quad \forall i \in M,
\end{aligned}$$

where π_j is the cut coefficient for variable x_j and w_i is a binary variable taking value 0 whenever the i -th row is considered in the disjunction (48). The inequality $\sum_{j=1}^n \pi_j x_j \geq 1$ separates x^* from \mathcal{D} if and only if $\eta < 0$. This separation problem is NP-hard since it is of the form of a version of the discrete probabilistic program with randomness only on right-hand sides, shown to be NP-hard in [82]. It is interesting that the simple “right-hand side randomness only” optimization problem studied in [82] arises as a separation problem for the “single row randomness” covering linear program, i.e., the portfolio optimization problem introduced in Section 4.1. Notice that although the mixing closure is contained in \mathcal{D} and separation over \mathcal{D} is \mathcal{NP} -complete, we do not know the complexity of the separation over $\text{Proj}_x(\bigcap_{\alpha} \text{conv}(Y(\alpha)))$.

2.5 Branching Scheme

As demonstrated in Table 1, the branch and bound search tree could be enormously large even for a small-sized instance of the MIP (10). Part of the reason for the excessive number of nodes is the overlap in the search tree. As one is not interested in solutions in which $z_j=1$ and $a_j^\top x \geq 1$, one does not exclude any interesting solutions by adding the constraint $a_j^\top x \leq 1$ on the $z_i = 1$ branch. This cut can also help trigger infeasibility-based pruning. Without loss of generality, we assume that z_j is the binary variable to branch on at the root node. The left branch with z_j fixed at zero consists

of the following set

$$\mathcal{B}^L := \{(x, z) : \sum_{i \neq j} z_i \leq k, a_j^\top x \geq 1, (x, z) \in X_{\text{MIP}}^j\},$$

where X_{MIP}^j represents the set X_{MIP} with the constraint $a_j^\top x + z_j \geq 1$ dropped and the variable z_j removed from the formulation. The right branch with z_j fixed at one consists of the following set

$$\mathcal{B}^R := \{(x, z) : \sum_{i \neq j} z_i \leq k - 1, a_j^\top x \geq 0, (x, z) \in X_{\text{MIP}}^j\},$$

which is the union of the following two sets:

$$\mathcal{B}_{\geq}^R := \{(x, z) : \sum_{i \neq j} z_i \leq k - 1, a_j^\top x \geq 1, (x, z) \in X_{\text{MIP}}^j\}$$

and

$$\mathcal{B}_{\leq}^R := \{(x, z) : \sum_{i \neq j} z_i \leq k - 1, a_j^\top x \leq 1, (x, z) \in X_{\text{MIP}}^j\}.$$

Note that \mathcal{B}_{\geq}^R is in fact a restriction of \mathcal{B}^L and hence an overlap between the left and right branches. Re-exploring \mathcal{B}_{\geq}^R in the right branch is a redundancy which could also hinder the infeasibility-based pruning: When \mathcal{B}_{\leq}^R is infeasible but \mathcal{B}_{\geq}^R is feasible, the overall right branch will be treated as a feasible node that, otherwise, would have been pruned. We can safely take \mathcal{B}_{\geq}^R out of the right branch by adding a local cut $a_j^\top x \leq 1$ and the remaining search tree will still cover the whole solution space. This logic applies to any node with a z variable fixed at one.

2.6 Computational Experiments

In this section, we examine the potential impact of the proposed MIP approaches in solving two classes of problems with the CKVLP structure, i.e. MIPs of the form of (10). We implement the algorithms using CPLEX callable libraries (version 12.1), run the programs on Intel Xeon 2.27 GHz dual core Linux servers installed with 4 Gb RAM, and compare the performance against the CPLEX MIP solver with default settings.

2.6.1 Implementation Details

The implementation of the coefficient strengthening technique (described in Section 2.3) straightforwardly follows Algorithm 1. Notice that, we could obtain a tighter ℓ^t by enforcing integrality constraints on some binary variables in $X(\ell^t)$, but the series of minimization problems in Algorithm 1 would become more time-consuming. We keep $X(\ell^t)$ in Algorithm 1 as the set in Definition 1. The threshold parameter Δ is chosen to be 0.001.

In the implementation of the mixing set inequality procedure (described in Section 2.4), we add cuts only at the root nodes of search trees. We first solve the root node LP relaxation and obtain an optimal solution (\bar{x}, \bar{z}) . Next we select the vector α from the following two sets:

- those constraint vectors a_i s for which $a_i^\top \bar{x} < 1$; and
- the cost vector c , if all a_i s have been used as α .

Notice that, in our implementation, we do not use formula (44) to calculate β_i^α directly. Instead, we add the LP relaxation of the original problem as additional constraints to generate the strongest inequalities, as suggested in [83]. Then we build a mixing set $Y(\alpha)$ as described in Section 2.4. Other than the most violated mixing inequality from (43), we also add violated inequalities (43) with $|T| = 2$ and $t_1 = 1$ to the root-node LP relaxation and solve it. The choice of these inequalities is based on recommendations in [83]. We iterate this process until one of the following stopping criteria is reached: (1) no cut with a violation of more than 0.00001 is identified, (2) the solution time exceeds 10,000 seconds, or (3) the cut generation procedure has run for 1000 iterations. To obtain the most violated mixing inequality, we implemented the separation algorithm in [13]. At the end of the cut generation phase, we keep only the tight cuts in the final model that is passed on to the branch-and-bound phase.

In the implementation of the branching rule, we add $a_i^\top x \leq 1$ as a local cut to the nodes in which z_i is fixed at one.

In addition, we also test the usefulness of the theoretical lower bound for ℓ^* in Lemma 2 by tightening the original formulation using $\ell_i = \frac{m-k}{m-\rho k} \rho_i$ directly and then solving the tightened formulation with the CPLEX MIP solver at default settings. We call the strengthened formulation by theoretical bounds as TB. Note that TB can be obtained by going through each entry of the left-hand-side matrix and calculating ρ_i and ρ using formula (21) and (22), respectively.

2.6.2 Probabilistic Portfolio Optimization

The first class of instances we test are from the probabilistically-constrained portfolio optimization model (8) introduced in Section 4.1. This problem can be approximated by the sample approximation approach as in (9) and reformulated as the following MIP [95]:

$$\begin{aligned}
\min \quad & c^\top x \\
\text{s.t.} \quad & e^\top x = 1 \\
& a_i^\top x + rz_i \geq r, \quad \forall i = 1, \dots, m \\
& \sum_{i=1}^m z_i \leq k, \\
& x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad \forall i = 1, \dots, m.
\end{aligned}$$

where a_i is the i -th sample drawn from the distribution of \tilde{a}_i and $k = \lfloor m \times \epsilon \rfloor$. The k -violation substructure in this formulation implies that the number of sampled scenarios in which the overall return is not achieved must not exceed $\lfloor m \times \epsilon \rfloor$. Hence, the frequency $\frac{k}{m}$ approximates the risk level ϵ . The constraint $e^\top x = 1$ is the budget constraint obtained by scaling the investment levels to a unit budget. We also considered instances in which there is no budget constraint.

Each component of a_i is drawn from an independent uniform distribution between

0.8 and 1.5, which, in this context, represents the range between a 20% loss on one's investment and a 50% profit. The required return r is chosen to be 1.1, and ϵ is set at 0.075, indicating a ten percent average return with a probability of 92.5%. We set $n = 20$, $m = 200$, and $k = 15$, allowing, at most, 15 of 200 linear inequalities to be violated. The cost coefficients in the model with a budget constraint take on integer values uniformly distributed between 1 and 100. For the model without the budget constraint, we use the vector with all components equal to one as the cost vector, since the instances with this particular cost vector are especially difficult to solve. We select ten randomly generated instances for each model that can be solved by CPLEX within ten hours, and compare the proposed methods against CPLEX with default settings.

Tables 2 and 3 present the computational results for the model with a budget constraint and the model without a budget constraint, respectively. The first column gives the instance number. The second and third columns give the branch-and-bound (B&B) time (in seconds) and nodes of the CPLEX MIP solver (CPX). Columns 4-7 give the root node gap closed by the cuts generated by CPX, the strengthened formulation with theoretical bounds (TB), the coefficient strengthening (CS), and the mixing set inequalities (MIX), respectively. Finally, columns 8-11 and 12-15 compare the percentage improvements of the four schemes: the branching rule (BR), TB, CS, and MIX, over the CPLEX MIP solver on branch-and-bound time and nodes, respectively. The percentage improvement in time for BR is computed as $100 \times (\text{Time}(\text{CPX}) - \text{Time}(\text{BR})) / \text{Time}(\text{CPX})$, where $\text{Time}(\text{CPX})$ is the branch-and-bound time for default CPLEX and $\text{Time}(\text{BR})$ is the branch-and-bound time using the proposed branching rule. The percentage improvements for the other three schemes, and the nodes saved are computed analogously.

The reported solution times are only for the branch-and-bound phase of the overall procedure. The mixing set cutting plane algorithm spends 20 to 30 seconds on root

node until no more cuts can be separated. The time spent on coefficient strengthening, which amounts to solving a series of linear programming problems, is under 20 seconds. The local cuts added in the branching scheme can be obtained instantly by simply reversing the sign of the corresponding constraint. Since the preprocessing times in these instances are negligible in comparison with the branch-and-bound times, we do not include them in the solution time.

Table 2: Percentage Improvements Over CPLEX (Portfolio Optimization with Budget Constraint)

Instance Number	CPX Default		Root Gap Closed				B&B Time Saved				B&B Nodes Saved			
	B&B time	B&B nodes	CPX	TB	CS	MIX	BR	TB	CS	MIX	BR	TB	CS	MIX
1	11094	13640260	19%	12%	24%	25%	91%	85%	89%	60%	89%	73%	85%	62%
2	968	1657606	19%	11%	23%	24%	75%	71%	82%	51%	70%	59%	79%	54%
3	223	505037	36%	19%	36%	41%	77%	77%	86%	25%	77%	74%	86%	36%
4	19830	25651409	7%	3%	10%	13%	90%	83%	91%	79%	88%	75%	89%	80%
5	400	786701	29%	17%	31%	33%	70%	71%	69%	-4%	66%	61%	62%	-3%
6	5044	9786835	14%	7%	19%	22%	68%	59%	82%	27%	66%	45%	80%	37%
7	10923	14365495	29%	20%	33%	35%	82%	69%	88%	26%	80%	52%	85%	35%
8	1822	3177889	12%	3%	18%	21%	87%	58%	90%	20%	86%	48%	88%	35%
9	2115	3454682	17%	7%	24%	28%	71%	65%	69%	-17%	66%	52%	65%	-2%
10	13017	10526548	15%	8%	18%	21%	75%	64%	88%	11%	79%	30%	76%	-15%
Average	6544	8355246	20%	11%	24%	26%	79%	70%	83%	28%	77%	57%	80%	32%

Table 3: Percentage Improvements Over CPLEX (Portfolio Optimization Without Budget Constraint)

Instance Number	CPX Default		Root Gap Closed				B&B Time Saved				B&B Nodes Saved			
	B&B time	B&B nodes	CPX	TB	CS	MIX	BR	TB	CS	MIX	BR	TB	CS	MIX
0	11,903	23,786,322	2%	48%	64%	57%	-15%	73%	85%	-176%	22%	72%	91%	85%
1	14,584	24,366,521	4%	50%	66%	61%	29%	72%	95%	65%	38%	82%	95%	89%
2	8,730	17,586,672	2%	48%	64%	58%	14%	67%	92%	-181%	19%	77%	93%	88%
3	5,516	10,898,121	5%	48%	64%	59%	7%	54%	90%	51%	19%	70%	91%	83%
4	12,462	18,021,273	4%	51%	66%	62%	19%	72%	95%	65%	19%	80%	94%	89%
5	21,475	30,948,921	2%	48%	64%	58%	58%	53%	92%	65%	46%	78%	93%	84%
6	6,928	14,634,688	2%	48%	64%	60%	-27%	70%	86%	44%	17%	71%	88%	80%
7	15,547	20,957,656	2%	50%	65%	61%	42%	72%	93%	68%	33%	81%	94%	89%
8	34,512	68,752,624	2%	49%	64%	55%	41%	58%	89%	63%	50%	81%	94%	84%
9	5,314	9,376,843	2%	49%	65%	60%	-2%	79%	94%	69%	14%	81%	94%	88%
Average	13,697	23,932,964	3%	49%	65%	59%	17%	67%	91%	13%	28%	77%	93%	86%

From Tables 2 and 3 we observe that the mixing set inequalities and coefficient strengthening have comparable performance in terms of closing root node gaps. They both close more gap than the CPLEX default cuts, especially in the model without a

budget constraint. However, in the branch-and-bound process afterwards, the mixing set inequalities cannot take full advantage of the tighter lower bounds to reduce overall time and nodes. In fact, in four of the 20 instances, the mixing set inequalities even worsen the performance. The reason lies in the difficulty of selecting effective cuts to keep in the model throughout the branch-and-bound process. In our experiment, we also try to employ the CPLEX cut pool to dynamically manage all the cuts generated at root nodes, but we have not been successful in identifying the most useful cuts.

The coefficient strengthening technique closes gap amounts similar to those closed by the mixing inequalities, but the improvement in the overall branch-and-bound process is significantly larger than for the mixing inequalities. The coefficient strengthening is able to cut down the time and nodes by an average of over 80%. This achievement can be attributed to the fact that the coefficient strengthening tightens the lower bound without introducing any extra variables or constraints at the root node.

Although TB is not as effective as applying the strengthening procedure iteratively (CS), both Tables 2 and 3 show that the theoretical lower bounds for ℓ^* have reasonably good quality, reducing 60-70% of the solution time and search tree size on average. If we compare Tables 2 and 3, we observe that TB is less effective than CS in closing root node gaps and reducing search tree nodes in the instances with the budget constraint than in the instances without the budget constraint. This might be because the lower bound in Lemma 2 does not consider any extra constraint, e.g., the budget constraint in this case. Therefore, u^{t+1} in (32) is weaker and so is the lower bound for ℓ^* .

The branching rule performs surprisingly better in the model with the budget constraint with over 70% savings in nodes and time than in the model without the budget constraint where the savings are less than 30%. This sizable difference can be explained by the presence of the budget constraint. The budget constraint, as

one type of side constraint, greatly reduces the feasible region of the node problems. Consequently, the feasibility of the node problems that have budget constraints is more sensitive to the addition of local cuts obtained by reversing the signs of the corresponding covering inequalities. Therefore, adding the local cuts to the models with budget constraints is more likely to lead to infeasible node problems, triggering the infeasibility-based node pruning more frequently.

We also experimented various combinations of CS, MIX, and BR together, but the additional improvements were not significant.

2.6.3 Optimal Vaccination Allocation

The second class of test instances is the optimal vaccination allocation problem under uncertainty addressed in [114]. The vaccination allocation problem is to allocate a scarce vaccine to households in a community to prevent an epidemic from breaking out. The epidemic will die out if the post-vaccination reproductive number is strictly less than one. Assume a community has a set F of types of households and each type of household $f \in F$ consists of a combination of person types $t \in T$, e.g., child, adult, or elderly. A vaccination policy $v \in V$ is a delivery of vaccine to certain types of persons in a household $f \in F$. For example, a vaccination policy could be a delivery of vaccine only to the two children in a household type that consists of two adults and two children. The decision problem is to determine an implementation of vaccination policies for each type of household in this community with a minimal cost which guarantees that the post-vaccination reproductive number is strictly below one with a high probability $1 - \epsilon$. We state below the probabilistically-constrained model in [114]:

$$\begin{aligned}
\min : & \sum_{f \in F} \sum_{v \in V} \sum_{t \in T} v_t h_f x_{fv} \\
\text{s.t.} & \sum_{v \in V} x_{fv} = 1 \quad \forall f \in F \\
& \mathbb{P}\left\{ \sum_{f \in F} \sum_{v \in V} a_{fv}(\omega) x_{fv} \leq 1 \right\} \geq 1 - \epsilon \\
& 0 \leq x_{fv} \leq 1 \quad \forall f \in F, v \in V,
\end{aligned}$$

where x_{fv} is the decision variable representing the percentage of policy v to be implemented for household type f , v_t is the number of people of type t vaccinated in policy v , h_f is the proportion of households in the community that are of type f , and $a_{fv}(\omega)$ is the computed random parameter for impact of the vaccination policy v for household type f , which is a function of different random numbers following some known distributions. For more details, see [16, 114].

After m i.i.d. samples are taken from $a_{fv}(\omega)$ s, the above probabilistically-constrained problem can be approximated by the following MIP, which has a CKVLP structure:

$$\begin{aligned}
\max : & \sum_{f \in F} \sum_{v \in V} \sum_{t \in T} v_t h_f x'_{fv} - \sum_{f \in F} \sum_{v \in V} \sum_{t \in T} v_t h_f \\
\text{s.t.} & \sum_{v \in V} x'_{fv} = |V| - 1 \quad \forall f \in F \\
& \sum_{f \in F} \sum_{v \in V} a_{fv}^i x'_{fv} + b_i z_i \geq b_i \quad i = 1, \dots, m \\
& \sum_{i=1}^m z_i \leq k \\
& 0 \leq x'_{fv} \leq 1 \quad \forall f \in F, v \in V, z_i \in \{0, 1\} \quad i = 1, \dots, m,
\end{aligned}$$

where a_{fv}^i is the i -th sample of $a_{fv}(\omega)$, $x'_{fv} = 1 - x_{fv}$, $b_i = \sum_{f \in F} \sum_{v \in V} a_{fv}^i - 1$, and $k = \lfloor \epsilon \times m \rfloor$.

We use the same test instances of this problem as in [114]. These instances have 302 continuous variables and m binary variables (see Column 1 in Table 4). The risk level ϵ is set to 0.05, and the value of k can be determined accordingly by $k = \lfloor m \times \epsilon \rfloor$.

Table 4 compares the performance of three schemes against the performance of the CPLEX MIP solver. The first two columns describe the sizes of the instances.

The next three columns provide the root node gaps closed by the cuts generated by the CPLEX MIP solver, the coefficient strengthening procedure, and the mixing inequalities, respectively. Columns 6-7 present the time (in seconds) spent on coefficient strengthening and generating mixing inequalities at the root node, respectively. Columns 8-11 and columns 12-15 compare the time (in seconds) and the number of nodes in the branch-and-bound phase by the CPLEX MIP solver and the three proposed schemes, respectively. Table 5 summarizes the percentage improvements of the three schemes over the CPLEX MIP solver with default settings. The percentage improvements in total time (root node time + branch-and-bound time) for CS is computed as $100 \times (\text{Time}(\text{CPX}) - \text{Time}(\text{CS})) / \text{Time}(\text{CPX})$, where $\text{Time}(\text{CPX})$ is the total time for default CPLEX and $\text{Time}(\text{CS})$ is the total time using coefficient strengthening. The percentage improvements in the branch-and-bound time (excluding the coefficient strengthening time) and the nodes saved are computed analogously. The percentage improvements for MIX and BR are computed similarly. We omitted the columns for TB since it is not helpful in this class of instances.

Table 4: Computational Results (Optimal Vaccination Allocation Problem)

Size m	k	Root Gap Closed			Root Node Time			B&B Time				B&B Nodes			
		CPX	CS	MIX	CS	MIX		CPX	CS	MIX	BR	CPX	CS	MIX	BR
250	12	62%	95%	95%	103	155		2	0	0	1	81	46	67	49
		61%	93%	92%	100	160		3	0	0	2	673	81	129	134
		64%	93%	93%	114	168		5	0	0	2	1,096	83	52	91
		58%	92%	90%	102	174		9	0	0	2	3,054	39	112	149
		55%	94%	94%	106	150		2	0	0	1	153	33	24	33
500	25	52%	91%	89%	493	1,776		42	4	6	13	7,082	286	820	995
		50%	93%	90%	507	1,738		64	6	14	14	20,166	1,004	2,600	1,551
		55%	91%	90%	532	1,832		28	2	5	11	5,329	218	390	839
		54%	90%	90%	515	1,731		22	2	4	11	3,907	347	359	694
		48%	94%	92%	506	1,964		50	1	2	9	18,822	38	26	458
750	37	30%	91%	90%	1,656	9,230		1,340	7	16	59	206,179	463	1,648	5,276
		44%	91%	91%	1,574	9,066		61	15	16	158	7,124	1,554	1,989	12,944
		49%	93%	92%	1,522	8,418		101	11	17	43	16,857	804	1,436	2,729
		45%	91%	90%	1,106	8,101		232	8	16	87	40,393	1,057	1,497	6,692
		42%	91%	90%	1,116	8,164		130	8	18	35	23,886	1,198	1,332	2,658
1000	50	26%	90%	89%	2,928	10,162		6,449	30	35	182	829,738	3,142	4,773	6,351
		32%	91%	89%	3,107	10,056		3,636	86	143	482	433,591	11,998	16,333	15,915
		29%	92%	90%	3,157	10,144		4,546	28	166	376	348,976	2,552	13,239	15,253
		38%	91%	88%	3,218	10,012		350	25	137	185	28,423	2,424	15,406	9,278
		33%	91%	88%	2,791	10,109		176	21	79	163	15,241	1,821	6,310	7,065
2000	100	16%	88%	84%	10,684	10,804		166,074	843	36,096	13,992	9,978,113	57,601	2,237,052	294,053
		15%	89%	85%	10,259	11,084		386,246	2,740	185,168	15,960	24,523,780	191,138	10,109,370	316,629
		14%	89%	86%	10,570	11,276		324,023	1,779	49,923	6,305	24,463,163	116,273	3,297,139	114,177
		15%	89%	85%	11,082	11,091		141,172	2,023	24,990	5,129	9,032,616	138,575	1,350,094	89,084
		15%	89%	84%	11,257	11,448		574,819	1,889	493,400	10,795	39,500,399	128,181	27,842,365	184,543

Table 5: Percentage Improvements Over CPLEX (Optimal Vaccination Allocation Problem)

Size		B&B Node Saved			B&B Time Saved			Total Time Saved		
m	k	CS	MIX	BR	CS	MIX	BR	CS	MIX	BR
250	12	43%	17%	40%	100%	100%	50%	-5033%	-7669%	50%
		88%	81%	80%	100%	100%	33%	-3220%	-5237%	33%
		92%	95%	92%	100%	100%	60%	-2186%	-3257%	60%
		99%	96%	95%	100%	100%	78%	-1037%	-1836%	78%
		78%	84%	78%	100%	100%	50%	-5191%	-7379%	50%
500	25	96%	88%	86%	90%	86%	69%	-1083%	-4142%	69%
		95%	87%	92%	91%	78%	78%	-701%	-2637%	78%
		96%	93%	84%	93%	82%	61%	-1808%	-6459%	61%
		91%	91%	82%	91%	82%	50%	-2251%	-7785%	50%
		100%	100%	98%	98%	96%	82%	-913%	-3833%	82%
750	37	100%	99%	97%	99%	99%	96%	-24%	-590%	96%
		78%	72%	-82%	75%	74%	-159%	-2506%	-14788%	-159%
		95%	91%	84%	89%	83%	57%	-1418%	-8252%	57%
		97%	96%	83%	97%	93%	63%	-380%	-3399%	63%
		95%	94%	89%	94%	86%	73%	-764%	-6194%	73%
1000	50	100%	99%	99%	100%	99%	97%	54%	-58%	97%
		97%	96%	96%	98%	96%	87%	12%	-180%	87%
		99%	96%	96%	99%	96%	92%	30%	-127%	92%
		91%	46%	67%	93%	61%	47%	-827%	-2800%	47%
		88%	59%	54%	88%	55%	7%	-1498%	-5689%	7%
2000	100	99%	78%	97%	99%	78%	92%	93%	72%	92%
		99%	59%	99%	99%	52%	96%	97%	49%	96%
		100%	87%	100%	99%	85%	98%	96%	81%	98%
		98%	85%	99%	99%	82%	96%	91%	74%	96%
		100%	30%	100%	100%	14%	98%	98%	12%	98%

The results in Table 4 and 5 show the effectiveness of the coefficient strengthening technique in both closing root node gaps and reducing nodes and time of the branch-and-bound phase. We observe that the performance of the coefficient strengthening algorithm is significantly more consistent than the other two methods and exhibits a certain stability. For example when $m = 1000$, the branch-and-bound time saved by the branching scheme ranges from 7.4% to 97.2%; the branch-and-bound time saved by the mixing set inequalities ranges from 55.1% to 99.5%; in contrast, the coefficient strengthening algorithm varies only from 88.1% to 99.5%. This consistent behavior is also observed for the probabilistic portfolio optimization instances in Tables 2 and 3. The branching scheme has a comparable impact on reducing the search tree size to the coefficient strengthening in the vaccination instances, especially for the difficult ones

with $m = 2000$. Since this model consists of equalities as side constraints, the local cuts added by the branching rule cause infeasibility in the node problems frequently, therefore, effectively reducing the search tree size.

The performance improvement in the branch-and-bound phase comes at the expense of computational effort in coefficient strengthening and separation of mixing inequalities at the root node. Unlike the portfolio optimization instances, this effort is quite significant for the vaccination instances (see columns 6-7 in Table 4). Each iteration of the coefficient strengthening requires solving m linear programs – for the instances with $m = 1000$ and $m = 2000$, several thousand linear programs need to be solved. Similarly, in generating the mixing set inequalities, m non-trivial linear programs, each of which consists of $O(m)$ constraints, need to be solved in order to form one mixing set for a given α , and there are m possible choices for α . Accordingly, the cut generation time increases in the order of m^2 . Comparing column 8 in Table 4 and column 9 in Table 5, we observe that significant effort on coefficient strengthening is not justifiable for instances that CPLEX can solve in under 1500 seconds. For example, for the instances with $m = 1000$, the coefficient strengthening technique takes around 3000 seconds. Recall that we impose a time limit of 10000 seconds, so for these instances coefficient strengthening is run till no coefficients can be further tightened. Considering the fact that CPLEX takes only one to two hours to solve these instances, running the strengthening procedure to termination is not economical. Similarly, we observe (by comparing column 8 in Table 4 and column 10 in Table 5) that the effort on mixing inequalities is not justified for instances with $m < 2000$ that CPLEX can solve within 6500 seconds. On the overall solution time, the branching rule has a more consistent performance since it requires no additional effort at the root node. For the larger size instances with $m = 2000$, it is worth spending about three hours on strengthening to reduce the branch-and-bound time

from days to minutes. The CPLEX MIP solver takes one to six days to solve these instances to optimality, whereas the coefficient strengthening reduces the overall effort to under four hours.

In our experiments, we focus on the maximum gap that could be closed by coefficient strengthening. Therefore, we run the procedure to the end, i.e., the improvement from the previous iteration is sufficiently small. However, the gap closed is not proportional to the number of iterations and the time spent on strengthening. In Figure (1), we illustrate how the LP optimal value changes over time for the vaccination instances with $m=1000$.

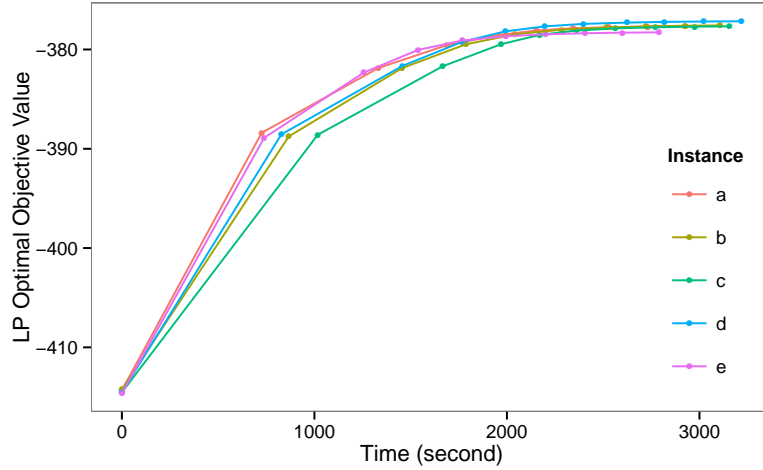


Figure 1: LP Bound Improvement Over Time/Iteration ($m = 1000$)

The x -axis represents accumulated time in seconds and the y -axis represents the LP optimal values. The LP optimal objective value is recorded after each iteration. The point at time zero represents the initial LP relaxation. We notice that most of the gap closing occurs during the first three to four iterations and then it begins to tail off. All other instances exhibit a similar curve for the strengthening process. To achieve a balance between tightness of the strengthened formulation and the effort spent on strengthening, we could terminate the strengthening procedure earlier, e.g., by setting the threshold parameter ϵ in Algorithm 1 to a larger value.

The mixing set also spends enormous time on generating cuts at the root node. In our implementation, each β_i is calculated by a linear program with $O(m)$ constraints, including the original LP relaxation. Therefore, the calculation is extremely time consuming when m becomes large, accounting for 99% of the time spent on root nodes. As pointed out in [83], there is a trade-off between spending more time improving coefficients and the resulting improvement in relaxation gap. By choosing a simpler linear program, i.e., dropping the constraints from the original LP relaxation, the time in calculating β_i can be greatly reduced. However, this could produce weaker mixing set inequalities. In our experiments, we have chosen to generate the strongest cuts to close as much root node gap as possible and observe its impact on the branch-and-bound process.

2.7 Concluding Remarks

In this chapter, we study covering-type k -violation linear programs. We show that such problems are strongly NP-hard, and study empirically the computational difficulty of MIP-based approaches for these problems. We introduce and analyze a coefficient strengthening scheme, adapt and analyze an existing cutting plane technique, and present a branching technique to improve the performance of MIP approaches. Computational experiments on two classes of problems show that the proposed methods are effective in significantly reducing running times. The coefficient strengthening is most effective for large instances and reduces the solution time and the number of search tree nodes by 80% to 98% in these instances. The branching scheme reduces the size of search trees by removing overlaps between branches and incurring infeasibility-based node pruning. It takes no effort to implement and works most effectively on the CKVLP models with side constraints. The mixing set cuts are capable of closing a large percentage of root node gaps. However, the impact of these cuts on the branch-and-bound process are mixed. Perhaps better performance might be

achieved by a more effective separation procedure for mixing inequalities. We have also investigated the performance of various combinations of the three schemes, but the gains are not significant.

CHAPTER III

CUTTING PLANES FOR PROBABILISTIC COVERING LINEAR PROGRAMS

3.1 *Introduction*

In this chapter, we continue to develop solution algorithms for solving MIP formulations of chance constrained covering linear programs defined as follows

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x + b_i z_i \geq b_i \quad i = 1, \dots, m, \end{aligned} \tag{50}$$

$$\sum_{i=1}^m z_i \leq k \tag{51}$$

$$x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad i = 1, \dots, m, \tag{52}$$

$$x \in X, \tag{53}$$

where $c \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$, and $b_i \in \mathbb{R}$ are all non-negative values, and X is a set defined by linear constraints, such as the budget constraint $e^\top x = 1$, where $e \in \mathbb{R}^n$ is a vector of ones, as in the portfolio optimization problems and vaccination allocation problems tested in Chapter 2 Section 6. We focus on the structure of the above set and develop cutting planes. Clearly, any subset of constraints in (50)-(53) define a relaxation of (50)-(53) and valid inequalities derived for the relaxations are also valid for the set defined by the constraints in (50)-(53). In the following sections, we study two types of typical relaxations and derive valid inequalities.

3.2 Valid Inequalities Induced by Resource Constraints

The first type of relaxation is formed by one of the constraints in (50) and the resource constraint $e^\top x = 1$.

$$S := \{(x, z) \in \mathbb{R}_+^n \times \{0, 1\} : \sum_{j=1}^n a_j x_j + bz \geq b, \sum_{j=1}^n x_j = 1\}. \quad (54)$$

Notice that the subscript i is dropped for simplicity. The resource constraint can take a more general form, e.g., $\sum_{j=1}^n d_j x_j = g$. In this case, we can scale the resource constraint and rewrite it as the standard form in (54):

$$S = \{(x', z) \in \mathbb{R}_+^n \times \{0, 1\} : \sum_{j=1}^n \frac{a_j}{d_j} x'_j + \frac{b_i}{g} z \geq \frac{b_i}{g}, \sum_{j=1}^n x'_j = 1\}, \quad (55)$$

where $x'_j = \frac{d_j}{g} x_j$.

Without loss of generality, we assume that a_1 is the smallest coefficient; and we eliminate x_1 in set S using the equality constraint $\sum_{j=1}^n x_j = 1$:

$$S = \{(x, z) \in \mathbb{R}_+^{n-1} \times \{0, 1\} : \sum_{j=2}^n (a_j - a_1)x_j + bz \geq b - a_1, \sum_{j=2}^n x_j \leq 1\}$$

Since $b \geq b - a_1$, we can reduce the coefficient of z to $b - a_1$. For sake of convenience, we reindex the above set and rewrite it as follows

$$S = \{(x, z) \in \mathbb{R}_+^n \times \{0, 1\} : \sum_{j=1}^n a_j x_j + a_0 z \geq a_0, \sum_{j=1}^n x_j \leq 1\}, \quad (56)$$

where $0 < a_1 < a_2 < \dots < a_k < a_0 \leq a_{k+1} < \dots < a_n$.

Proposition 11.

$$\begin{aligned} \text{conv}(S) = \{ & (x, z) \in \mathbb{R}_+^n \times \mathbb{R}_+ : \\ & \sum_{j=i+1}^n (a_j - a_i)x_j + (a_0 - a_i)z \geq a_0 - a_i \quad \forall i \in \{1, \dots, k\} \\ & \sum_{j=1}^n a_j x_j + a_0 z \geq a_0 \\ & \sum_{j=1}^n x_j \leq 1 \\ & z \leq 1\}. \end{aligned}$$

Proof. Since there is only one binary variable, we first consider the following extended formulation.

$$\begin{aligned}
\sum_{j=1}^n a_j x_j^1 + a_0 z^1 &\geq a_0 \lambda \\
\sum_{j=1}^n x_j^1 &\leq \lambda \\
z^1 &\leq \lambda \\
z^1 &\leq 0 \\
\sum_{j=1}^n a_j x_j^2 + a_0 z^2 &\geq a_0 (1 - \lambda) \\
\sum_{j=1}^n x_j^2 &\leq (1 - \lambda) \\
z^2 &\leq 1 - \lambda \\
z^2 &\geq 1 - \lambda \\
x_j^t &\geq 0 \quad \forall j \in \{1, \dots, n\}, \quad t \in \{1, 2\} \\
z^1, z^2 &\geq 0 \\
z &= z^1 + z^2 \\
x_j &= x_j^1 + x_j^2 \quad \forall j \in \{1, \dots, n\} \\
0 &\leq \lambda \leq 1.
\end{aligned}$$

The projection of the above set on to the space of variables (x, z) is the convex hull of X . First we observe that $z^1 = 0$, $z^2 = 1 - \lambda$, and $\lambda = 1 - z$. Therefore, after projecting out z^1, z^2 , and λ , we obtain the set,

$$\begin{aligned}
\sum_{j=1}^n a_j x_j^1 &\geq a_0 (1 - z) \\
\sum_{j=1}^n x_j^1 &\leq (1 - z) \\
\sum_{j=1}^n x_j^2 &\leq z \\
x_j &= x_j^1 + x_j^2 \quad \forall j \in \{1, \dots, n\} \\
x_j^t &\geq 0 \quad \forall j \in \{1, \dots, n\}, \quad t \in \{1, 2\} \\
0 &\leq z \leq 1.
\end{aligned}$$

Now we sequentially project out the auxiliary variables x_j^t $t = 1, 2$ and $j = 1, \dots, n$.

On projecting out x_1^2 , we obtain the set

$$\begin{aligned}
\sum_{j=1}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=1}^n x_j^1 &\leq (1-z) \\
\sum_{j=2}^n x_j^2 &\leq z + (x_1^1 - x_1) \\
x_j &= x_j^1 + x_j^2 \quad \forall j \in \{2, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{1, \dots, n\} \\
x_j^2 &\geq 0 \quad \forall j \in \{2, \dots, n\} \\
x_1 - x_1^1 &\geq 0 \\
0 &\leq z \leq 1.
\end{aligned}$$

(We additionally obtain the inequality $\sum_{j=2}^n x_j^2 \leq z$, implied by the above system.)

By repeating the same projection on variables x_1^2, \dots, x_l^2 , we get the set

$$\begin{aligned}
\sum_{j=1}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=1}^n x_j^1 &\leq (1-z) \\
\sum_{j=l+1}^n x_j^2 &\leq z + \sum_{j=1}^l (x_j^1 - x_j) \\
x_j &= x_j^1 + x_j^2 \quad \forall j \in \{l+1, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{1, \dots, n\} \\
x_j^2 &\geq 0 \quad \forall j \in \{l+1, \dots, n\} \\
x_j - x_j^1 &\geq 0 \quad \forall j \in \{1, \dots, l\} \\
0 &\leq z \leq 1.
\end{aligned}$$

Therefore, in particular, after projecting out variables x_1^2, \dots, x_n^2 , we obtain the set

$$\begin{aligned}
\sum_{j=1}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=1}^n x_j^1 &\leq (1-z) \\
\sum_{j=1}^n (x_j - x_j^1) &\leq z \\
x_j - x_j^1 &\geq 0 \quad \forall j \in \{1, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{1, \dots, n\} \\
0 &\leq z \leq 1.
\end{aligned}$$

On projecting out the variable x_1^1 , we obtain the set

$$\begin{aligned}
a_1 x_1 + \sum_{j=2}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=2}^n (a_j - a_1) x_j^1 &\geq (a_0 - a_1)(1-z) \\
\sum_{j=2}^n x_j^1 &\leq (1-z) \\
\sum_{j=2}^n (x_j - x_j^1) &\leq z \\
x_j - x_j^1 &\geq 0 \quad \forall j \in \{2, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{2, \dots, n\} \\
x_2 &\geq 0 \\
0 &\leq z \leq 1.
\end{aligned}$$

Projecting out x_1^1, \dots, x_l^1 (where $l \leq k$) in the same way, we have

$$\begin{aligned}
\sum_{j=1}^l a_j x_j + \sum_{j=l+1}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=i+1}^n (a_j - a_i) x_j^1 &\geq (a_0 - a_i)(1-z) \quad \forall i \in \{1, \dots, l\} \\
\sum_{j=l+1}^n x_j^1 &\leq (1-z) \\
\sum_{j=l+1}^n (x_j - x_j^1) &\leq z \\
x_j - x_j^1 &\geq 0 \quad \forall j \in \{l+1, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{l+1, \dots, n\} \\
x_j &\geq 0 \quad \forall j \in \{1, \dots, l\} \\
0 &\leq z \leq 1.
\end{aligned}$$

Thus, by projecting out x_1^1, \dots, x_k^1 , we obtain the set

$$\begin{aligned}
\sum_{j=1}^k a_j x_j + \sum_{j=k+1}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=i+1}^n (a_j - a_i) x_j^1 &\geq (a_0 - a_i)(1-z) \quad \forall i \in \{1, \dots, k\} \\
\sum_{j=k+1}^n x_j^1 &\leq (1-z) \\
\sum_{j=k+1}^n (x_j - x_j^1) &\leq z \\
x_j - x_j^1 &\geq 0 \quad \forall j \in \{k+1, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{k+1, \dots, n\} \\
x_j &\geq 0 \quad \forall j \in \{1, \dots, k\} \\
0 &\leq z \leq 1.
\end{aligned}$$

Now we project out x_{k+1}^1 and obtain the set

$$\begin{aligned}
\sum_{j=1}^k a_j x_j + \sum_{j=k+1}^n a_j x_j^1 &\geq a_0(1-z) \\
\sum_{j=i+1}^n (a_j - a_i) x_j^1 &\geq (a_0 - a_i)(1-z) \quad \forall i \in \{1, \dots, k\} \\
\sum_{j=k+2}^n x_j^1 &\leq (1-z) \\
\sum_{j=k+2}^n (x_j - x_j^1) &\leq z \\
x_j - x_j^1 &\geq 0 \quad \forall j \in \{k+2, \dots, n\} \\
x_j^1 &\geq 0 \quad \forall j \in \{k+2, \dots, n\} \\
x_j &\geq 0 \quad \forall j \in \{1, \dots, k+1\} \\
0 &\leq z \leq 1.
\end{aligned}$$

(since $a_{k+1} - a_0 \geq 0$, we get one less inequality.) Finally, on projecting out x_{k+2}^1, \dots, x_n^1 , we obtain the set

$$\begin{aligned}
\sum_{j=i+1}^n (a_j - a_i) a_j + (a_0 - a_i) z &\geq a_0 - a_i \quad \forall i \in \{1, \dots, k\} \\
\sum_{j=1}^n a_j x_j + a_0 z &\geq a_0 \\
\sum_{j=1}^n x_j &\leq 1 \\
0 &\leq z \leq 1 \\
x_j &\geq 0 \quad \forall j \in \{1, \dots, n\}.
\end{aligned}$$

□

Each of the m constraints in (50) has one binary variable. Therefore, we can have m relaxations of the form S and obtain no more than $n \times m$ nontrivial valid inequalities as shown in Proposition 11.

3.3 Valid Inequalities From Two Row Constraints

Relaxation also arises from two rows of constraints in (50), each of which consists of one binary variable:

$$P := \{(x, z_{i_1}, z_{i_2}) \in \mathbb{R}_+^n \times \{0, 1\} \times \{0, 1\} : a_{i_1}^\top x + b_{i_1} z_{i_1} \geq b_{i_1}, a_{i_2}^\top x + b_{i_2} z_{i_2} \geq b_{i_2}\}. \quad (57)$$

Since $b_{i_1} > 0$ and $b_{i_2} > 0$, we can normalize the right-hand sides and rewrite P with a simplified notation:

$$P = \{(x, z_1, z_2) \in \mathbb{R}_+^n \times \{0, 1\} \times \{0, 1\} : a^\top x + z_1 \geq 1, b^\top x + z_2 \geq 1\}. \quad (58)$$

To obtain valid inequalities for P , we first find valid inequalities for $P \cap \{z_2 = 0\}$, denoted as $P_{z_2=0}$, and then lift the inequalities into (x, z_1, z_2) -space. Assume that the valid inequalities for $P_{z_2=0}$ are of the form

$$\alpha^\top x + pz_1 \geq \beta,$$

and the lifted inequality has the following form: $\alpha^\top x + pz_1 + qz_2 \geq \beta$. The lifted coefficient q can be obtained by solving the following optimization problem:

$$\begin{aligned} \max \quad & q \\ q = \beta - & \sum_{i=1}^n \alpha_i x_i + pz_1 \\ \sum_{i=1}^n & a_i x_i + z_1 \geq 1 \\ x \geq 0, z_1 \in & \{0, 1\}. \end{aligned} \quad (59)$$

The optimal objective value of the optimization problem can be written in a closed form:

$$q = \beta - \min\left\{\frac{\alpha_1}{a_1}, \frac{\alpha_2}{a_2} \dots \frac{\alpha_n}{a_n}, p\right\}.$$

For simplicity, we drop the subscript of z_1 and define $P_{z_2=0}$ as follows

$$P_{z_2=0} := \{(x, z) \in \mathbb{R}_+^n \times \{0, 1\} : \sum_{i=1}^n a_i x_i + z \geq 1, \sum_{i=1}^n b_i x_i \geq 1\}.$$

Without loss of generality, we assume that $\frac{a_1}{b_1} > \dots > \frac{a_{\ell-1}}{b_{\ell-1}} > 1 > \frac{a_\ell}{b_\ell} > \dots > \frac{a_n}{b_n}$.

Proposition 12. *The following inequalities are the only non-trivial facet-defining inequalities of $P_{z_2=0}$:*

$$b_t \sum_{i=1}^t a_i x_i + a_t \sum_{i=t+1}^n b_i x_i + (b_t - a_t)z \geq b_t \quad t = \ell, \dots, n. \quad (60)$$

Proof. Since there is only one binary variable in set $P_{z_2=0}$, all valid inequalities for $P_{z_2=0}$ can be lifted from $P_{z_2=0} \cap \{z_1 = 0\}$. Let

$$Q := \{ (x, z) \in \mathbb{R}^n \times \{0, 1\} : \sum_{i=1}^n a_i x_i \geq 1 \} \quad (61)$$

$$\sum_{i=1}^n b_i x_i \geq 1 \quad (62)$$

$$x_i \geq 0 \quad i = 1, \dots, n \quad (63)$$

$$z = 0 \}$$

and

$$R := \{ (x, z) \in \mathbb{R}_+^n \times \{0, 1\} : \sum_{i=1}^n b_i x_i \geq 1, x_i \geq 0 \quad i = 1, \dots, n, \quad z = 1 \}.$$

Clearly, a facet defining inequality of $P_{z_2=0}$ supports extreme points of both Q and R . Since Q and R are simple polyhedrons, we can enumerate their extreme points. The extreme points of Q are $E_a := \{(e_i \frac{1}{a_i}, 0) \mid i = \ell, \dots, n\}$, $E_b := \{(e_i \frac{1}{b_i}, 0) \mid i = 1, \dots, \ell - 1\}$, and $E_{ab} = \{(e_i \frac{a_j - b_j}{a_j b_i - a_i b_j} + e_j \frac{b_i - a_i}{a_j b_i - a_i b_j}, 0) \mid i < \ell, j \geq \ell\}$, where e_i is the i -th unit vector. The set of extreme points of R is $E_R = \{(e_i \frac{1}{b_i}, 1) \mid i = 1, \dots, n\}$.

Let the non-negative multipliers associated with (61), (62), and (63) be λ , μ , and $\pi_i \mid i = 1, \dots, n$, respectively. Then any valid inequality for Q can be written in the following form:

$$\sum_{i=1}^n (\lambda a_i + \mu b_i + \pi_i) x_i \geq \lambda + \mu. \quad (64)$$

Let the inequality in (x, z) -space after lifting be

$$\sum_{i=1}^n (\lambda a_i + \mu b_i + \pi_i) x_i + p z \geq \lambda + \mu, \quad (65)$$

where p is the lifted coefficient for z , which can be obtained by solving the following

lifting problem

$$\begin{aligned}
& \max \quad p \\
& \text{s.t. } p = \lambda + \mu - \sum_{i=1}^n (\lambda a_i + \mu b_i + \pi_i) x_i \\
& \quad \sum_{i=1}^n b_i x_i \geq 1 \\
& \quad x_i \geq 0 \quad \forall i = 1, \dots, n.
\end{aligned} \tag{66}$$

Notice that a non-trivial coefficient p should be positive. The optimal objective function value can be written in a closed form:

$$p = \mu + \lambda - \min_{i=1, \dots, n} \left(\lambda \frac{a_i}{b_i} + \mu + \frac{\pi_i}{b_i} \right).$$

Then the lifted inequality (65) has the following form:

$$\sum_{i=1}^n (\lambda a_i + \mu b_i + \pi_i) x_i + \left(\lambda + \mu - \min_{i=1, \dots, n} \left(\lambda \frac{a_i}{b_i} + \mu + \frac{\pi_i}{b_i} \right) \right) z \geq \lambda + \mu, \tag{67}$$

which we call as hyperplane H .

To be a facet-defining inequality, H needs to support $n + 1$ affinely independent points of $P_{z_2=0}$.

We discuss the following four cases with λ and μ taking different values and investigate the situation where H can support $n + 1$ affinely independent points.

1. $\lambda = 0$ and $\mu = 0$:

H becomes

$$\sum_{i=1}^n \pi_i x_i \geq \min_{i=1, \dots, n} \frac{\pi_i}{b_i} z. \tag{68}$$

- (a) If $\pi_i = 0$ for some i , the inequality in (68) becomes $\sum_{i=1}^n \pi_i x_i \geq 0$, which is a trivial inequality implied by the non-negativity of x_i ;
- (b) If $\pi_i > 0$ for all i , when $z = 0$, inequality (68) becomes $\sum_{i=1}^n \pi_i x_i \geq 0$, which does not support any extreme points of Q .

Therefore, the lifting problem in this case does not yield any non-trivial facet-defining inequality.

2. $\lambda = 0$ and $\mu > 0$:

The lifting problem yields $p = -\frac{\pi_i}{b_i} \leq 0$, therefore, there do not exist any non-trivial facet-defining inequalities.

3. $\lambda > 0$ and $\mu = 0$:

We show that in this case facet-defining inequalities can be obtained by setting the multipliers, λ , μ , and π_i , to appropriate values. Let E_H denote the extreme points of $P_{z_2=0}$ supported by H . Since $\mu = 0$, $E_H \cap E_b = \emptyset$. Since $\lambda > 0$, $E_H \cap E_a \neq \emptyset$. Consider an $x^* \in E_a \cap E_H$, where $x^* = (e_j \frac{1}{a_j}, 0)$ for some $j \geq \ell$. Since x^* is supported by H , we have

$$\frac{(\lambda a_j + \pi_j)}{a_j} = \lambda,$$

which implies that $\pi_j = 0$.

Let $\hat{x} \in E_H \cap E_R$, where $\hat{x} = (e_i \frac{1}{b_i}, 1)$ for some i . Since \hat{x} is supported by H , we have

$$\frac{\lambda a_i + \pi_i}{b_i} \geq \min_{i=1, \dots, n} (\lambda \frac{a_i}{b_i} + \frac{\pi_i}{b_i}).$$

Since $\lambda \frac{a_1}{b_1} > \dots > \lambda \frac{a_{j-1}}{b_{j-1}} > \lambda \frac{a_j}{b_j} \geq \min_{i=1 \dots n} (\lambda \frac{a_i}{b_i} + \frac{\pi_i}{b_i})$, points $(e_i \frac{1}{b_i}, 1)$ $i = 1, \dots, j-1$ are not supported by H . But points $(e_i \frac{1}{b_i}, 1)$ $i = j, \dots, n$ can be supported if we set π_i such that

$$\frac{\lambda a_i + \pi_i}{b_i} = \min_{i=1, \dots, n} (\lambda \frac{a_i}{b_i} + \frac{\pi_i}{b_i}) = \lambda \frac{a_j}{b_j} \quad i = j, \dots, n,$$

i.e., $\pi_j = 0$ and $\pi_i = \lambda \frac{a_j b_i - b_j a_i}{b_j} > 0$ for $i = j+1, \dots, n$. Therefore, when $z = 1$, we have $n - j + 1$ affinely independent points supported by H : $(e_i \frac{1}{b_i}, 1)$ $i = j, \dots, n$. When $z = 0$, we can set $\pi_i = 0$ for $i = 1, \dots, j$ and have points $(e_i \frac{1}{a_i}, 0)$ for $i = 1, \dots, j$ supported by H . Notice that $(e_i \frac{1}{a_i}, 0)$ for $i = 1, \dots, \ell - 1$

are not feasible points. However, by Lemma 1, each of these infeasible points can induce $j - \ell + 1$ feasible points in E_{ab} , which are also supported by H , by taking a convex combination of itself and the points $(e_i \frac{1}{a_i}, 0)$ for some $i \geq j$; there are j affinely-independent points among these $(\ell - 1)(j - \ell + 1)$ feasible points.

Thus, we have $n + 1$ feasible points supported by H and they are affinely independent. Plugging $\pi_i = 0$ for $i = 1, \dots, j$, $\pi_i = \lambda \frac{a_j b_i - b_j a_i}{b_j}$ for $i = j + 1, \dots, n$, and $\min_{i=1, \dots, n} (\lambda \frac{a_i}{b_i} + \frac{\pi_i}{b_i}) = \lambda \frac{a_j}{b_j}$ into (67), we obtain the following facet-defining inequality for $P_{z_2=0}$:

$$\sum_{i=1}^j a_i b_j x_i + \sum_{i=j+1}^n a_j b_i x_i + (b_j - a_j)z \geq b_j.$$

Since $|E_a| = n - \ell + 1$, we can obtain at most $n - \ell + 1$ facet-defining inequalities in this case, i.e., $j = \ell, \dots, n$.

4. $\lambda > 0$ and $\mu > 0$:

We discuss in the following three sub-cases and show that we cannot find $n + 1$ affinely independent points supported by H .

(a) $E_H \cap E_a \neq \emptyset$

Suppose $x^* \in E_a$ is supported by H , where $x^* = (e_j \frac{1}{a_j}, 0)$ for some $j \geq \ell$.

We plug x^* into (67):

$$\begin{aligned} \frac{(\lambda a_j + \mu b_j + \pi_j)}{a_j} &= \lambda + \mu, \\ \Rightarrow \pi_j &= \mu(a_j - b_j). \end{aligned}$$

Since $\mu > 0$ and $\pi_j \geq 0$, we have $a_j \geq b_j$, which contradicts the fact that $j \geq \ell$.

(b) $E_H \cap E_b \neq \emptyset$

Suppose $x^* \in E_b$ is supported by H , where $x^* = (e_j \frac{1}{b_j}, 0)$ for some $j < \ell$.

We plug x^* into (67):

$$\begin{aligned} \frac{(\lambda a_j + \mu b_j + \pi_j)}{b_j} &= \lambda + \mu, \\ \Rightarrow \pi_j &= \lambda(b_j - a_j). \end{aligned}$$

Since $\lambda > 0$ and $\pi_j \geq 0$, we have $b_j \geq a_j$, which contradicts the fact that $j < \ell$.

(c) $(E_H \cap \{z = 0\}) \subseteq E_{ab}$

Let $x^* \in E_{ab}$, where $x^* = (e_i \frac{a_j - b_j}{a_j b_i - a_i b_j} + e_j \frac{b_i - a_i}{a_j b_i - a_i b_j}, 0)$ for some $i < \ell$ and $j \geq \ell$. Suppose x^* is supported by H , plugging x^* into (67) we have

$$\begin{aligned} (\lambda a_i + \mu b_i + \pi_i) \frac{a_j - b_j}{a_j b_i - a_i b_j} + (\lambda a_j + \mu b_j + \pi_j) \frac{b_i - a_i}{a_j b_i - a_i b_j} &= \lambda + \mu \\ \Rightarrow \pi_i(a_j - b_j) + \pi_j(b_i - a_i) &= 0. \end{aligned}$$

Since $a_j - b_j < 0$ and $b_i - a_i < 0$, $\pi_i = \pi_j = 0$. Let $j^* := \max_i \{\pi_i = 0\}$. Then there are maximal possible $(\ell - 1)(j^* - \ell + 1)$ points in E_{ab} supported by H (when $\pi_i = 0$ for $i = 1, \dots, j^*$), of which there are only $j^* - 1$ affinely independent points (see Lemma 2). Let $\hat{x} \in E_R$, where $\hat{x} = (e_t \frac{1}{b_t}, 1)$ for some t . Suppose \hat{x} is supported by H , then we plug \hat{x} into (67) and arrive at

$$\lambda \frac{a_t}{b_t} + \frac{\pi_t}{b_t} = \min_{i=1, \dots, n} \left\{ \lambda \frac{a_i}{b_i} + \frac{\pi_i}{b_i} \right\}.$$

Since

$$\lambda \frac{a_t}{b_t} + \frac{\pi_t}{b_t} > \lambda \frac{a_{j^*}}{b_{j^*}} \geq \min_{i=1, \dots, n} \left\{ \lambda \frac{a_i}{b_i} + \frac{\pi_i}{b_i} \right\} \quad \forall t < j^*,$$

The points \hat{x} that are supported by H can only be $(e_t \frac{1}{b_t}, 1)$ for $t = j^*, \dots, n$. These $n - j^* + 1$ \hat{x} are affinely independent. Therefore, there are at most n affinely independent points in total that are supported by H and hence H cannot be a facet-defining inequality in this case.

We summarize that (60) are the only non-trivial facet-defining inequalities for $P_{z_2=0}$. □

Using the lifting problem in (59), we obtain the following result:

Proposition 13. *The following inequalities are valid for (50)-(53)*

$$b_t \sum_{i=1}^t a_i x_i + a_t \sum_{i=t+1}^n b_i x_i + (b_t - a_t) z_1 + q_t z_2 \geq b_t \quad t = \ell, \dots, n,$$

where $q_t = b_t - \min\{b_t, a_t \frac{b_{t+1}}{a_{t+1}}, a_t \frac{b_{t+2}}{a_{t+2}}, \dots, a_t \frac{b_n}{a_n}, b_t - a_t\}$.

Lemma 3. *When $z = 0$, let E_I be the set of intersection points by $\sum_{i=1}^n a_i x_i \geq 1$ and the non-negative orthant and $E_I \cap E_a = \emptyset$, i.e., $E_I = \{(e_i \frac{1}{a_i}, 0) \mid i = 1, \dots, \ell - 1\}$. Let $\hat{x} = (e_j \frac{1}{a_j}, 0) \in E_a$. If H supports \hat{x} and E_I , then H also supports $\ell - 1$ points from E_{ab} ; and all these points span an ℓ -dimensional space.*

Proof. Let $\hat{y} \in E_I$, i.e., $\hat{y} = (e_i \frac{1}{a_i}, 0)$ for some $i < \ell$. Since \hat{y} and \hat{x} share n linearly independent binding constraints, i.e., $x_t = 0$ for $t \in \{1, \dots, n\} \setminus \{i, j\}$, $\sum_{i=1}^n a_i x_i + z \geq 1$, and $z = 0$, \hat{x} and \hat{y} are supported by an n -dimensional face, which is a line in \mathbb{R}^{n+1} . Since \hat{y} is an infeasible point, violating the constraint $\sum_{i=1}^n b_i x_i \geq 1$, and \hat{x} is a feasible point, the line intersects the hyperplane $\sum_{i=1}^n b_i x_i \geq 1$ at a point w . Therefore, $w \in E_{ab}$ and can be written as a convex combination of \hat{x} and \hat{y} . Since $|E_I| = \ell - 1$, we can find $\ell - 1$ points in E_{ab} that are also supported by H . Because each w is a convex combination of a point in E_I and \bar{x} , the dimension of the space affinely spanned by all these points, w s, \bar{y} s, and \bar{x} , is determined by the dimension of the affine span of the set $E_I \cup \{\hat{x}\}$, which is $\ell - 1$. □

Lemma 4. *Let $W = \{(w^t, 0) \mid t = 1, \dots, q\}$ be a subset of E_{ab} such that the $w_i^t = 0$ for $i = j^* + 1, \dots, n$. Then there is at most $j^* - 1$ affinely independent points in W .*

Proof. Since all the points in W share the following $n - j + 3$ binding constraints: $\sum_{i=1}^n a_i x_i + z \geq 1$, $\sum_{i=1}^n b_i x_i \geq 1$, $z = 0$, and $x_i \geq 0$ for $i = j^* + 1, \dots, n$, W lies in a

$n + 1 - (n - j^* + 3) = j^* - 2$ dimensional space. Therefore, there are at most $j^* - 1$ affinely independent points in W . \square

3.4 Computational Experiments

The coefficient strengthening procedure proposed in Chapter 2 was demonstrated to be significantly helpful in solving the mixed-integer formulation of probabilistic covering linear programs as in (50)-(53). In this chapter, we incorporate the proposed valid inequalities with the coefficient strengthening procedure. The coefficient strengthening procedure tightens the coefficients on the binary variables in the following way:

1. Find a lower bound ℓ_i for each $a_i^\top x$ over a reasonable relaxation of the original feasible set, e.g., the linear programming relaxation of (50)-(53).
2. Tighten the coefficient for z_i using ℓ_i : $a_i^\top x + (b_i - \ell_i)z_i \geq b_i$.

The strengthened formulation after iteration t is as follows

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & a_i^\top x + (1 - \ell_i^t)z_i \geq 1 \quad i = 1, \dots, m \end{aligned} \tag{69}$$

$$\sum_{j=1}^n x_j = 1 \tag{70}$$

$$\sum_{i=1}^m z_i \leq k \tag{71}$$

$$x \in \mathbb{R}_+^n, z_i \in \{0, 1\} \quad i = 1, \dots, m. \tag{72}$$

The tightening procedure is iteratively carried out until no noticeable improvements on the LP lower bound. Let $S^t(x, z)$ be the set described with the constraints in (69)-(72) and $R^t(x, z)$ be the LP relaxation of $S^t(x, z)$, i.e., with the integrality constraints on z_i s dropped. Clearly, lower bound $\ell_i^0 = 0$ for $i = 1, \dots, m$, and $S^0(x, z)$ represents the original formulation. By Proposition 2 in Chapter 2, $S^0(x, z) = S^1(x, z) = \dots = S^t(x, z) \subseteq R^t(x, z) \subseteq \dots R^1(x, z) \subseteq R^0(x, z)$. Therefore, the inequalities derived by

using $S^0(x, z)$ are also valid for $S^t(x, z)$ for $t = 1, 2, \dots$. Hence, we generate the valid inequalities developed in this Chapter using the constraints in defining $S^0(x, z)$ and add them to $R^t(x, z)$ $t = 0, 1, \dots$, aiming to tighten these LP relaxations and obtain stronger lower bounds ℓ_i^t for $i = 1, \dots, m$ and $t = 1, 2, \dots$. With slight abuse of notations, we redefine $R^t(x, z)$ as the set defined by constraints (69)-(72) with integrality constraints dropped and the valid inequalities we derive by using $S^0(x, z)$.

In addition, we modify the coefficient strengthening procedure using the k -violation sub-structure implied in formulation (50)-(53) to further improve the lower bounds ℓ_i^t .

Proposition 14 ([82]). *Let $\beta_q(a_i) := \min\{a_i^\top x : a_q^\top x \geq 1, x \in R^{t-1}(x, z)\}$. W.l.o.g., assuming $\beta_1(a_i) > \dots > \beta_n(a_i)$, then $\ell_i^t = \beta_{k+1}(a_i)$ is a valid lower bound for $a_i x$ over the set $S^{t-1}(x, z)$.*

Since ℓ_i^t in Proposition 14 is determined by solving m LPs each of which consists of one more constraint than the LP in Algorithm 1 of Chapter 2, ℓ_i^t obtained by Proposition 14 is at least as good as the one obtained by Algorithm 1 of Chapter 2.

The implementation follows the framework of Algorithm 1 Iterative Coefficient Strengthening in Chapter 2. Since we do not have separation algorithms, we add valid inequalities to LP relaxation $S^t(x, z)$ $t = 0, 1, \dots$ prior to computation. We add all valid inequalities in Proposition 1. For the inequalities (60), since there are a large number $\binom{n}{2}$ of choices of relaxations to derive valid inequalities, we randomly select 100 pairs of the constraints in (50) to generate valid inequalities. After the strengthening procedure is done, we solve the strengthened model with CPLEX MIP solver with default settings.

The instance tested is a portfolio optimization problem with budget constraint introduced in Chapter 2, with $n = 20$, $m = 300$, and $k = 30$, which is also listed in MIPLIB 2010 [74]. The program was implemented with IBM ILOG CPLEX 12.2

callable libraries. The code was run on a Linux server with Dual Xeon E5520 quad-core processors at 2.27GHz installed with 32 GB memory. The CPLEX MIP solver was single threaded at its default settings. We list the solution reports by CPLEX MIP Solver at default settings (CPLEX), the coefficient strengthening technique introduced in Chapter 2 (CS), and the strengthened coefficient strengthening procedure developed in this chapter (SCS) in Table 3.4.

Table 6: Comparison of Performance on Solving a MIPLIB Instance

	Time Before Termination	Gap Remained	Status
CPLEX	381,455	10.71%	Out of Memory
CS	373,397	6.54%	Out of Memory
SCS	396,002	0.01%	Normal Termination

With the proposed strengthening algorithm, CPLEX solved the instance in 396,002 seconds with an optimal value 16.7342 within a 0.01% gap tolerance. The time spent on the strengthening procedure was 10,959 seconds.

CHAPTER IV

PROBABILISTIC K -COVER PROBLEMS

4.1 *Introduction*

Given a ground set $M := \{1, 2, \dots, m\}$ and subsets $M_j \subseteq M$ for all $j \in N := \{1, 2, \dots, n\}$, the set cover problem is to seek a minimum cost set of subsets whose union covers M , i.e., each element in the ground set is covered by at least one subset in the union [93, 117]. A natural extension of the classic set cover problem is the *set multi-cover problem*, or the *set k -cover problem*, in which each element needs to be covered at least k times. An integer programming formulation of the set multi-cover problem is

$$\begin{aligned} \min \quad & \sum_{j \in N} c_j x_j \\ & \sum_{j \in N} a_{ij} x_j \geq k_i \quad \forall i \in M, \\ & x_j \in \{0, 1\} \quad \forall j \in N, \end{aligned} \tag{73}$$

where $a_{ij} = 1$ if element $i \in M_j$; $a_{ij} = 0$, otherwise, and $k_i \geq 1$ for each i . c_j is the cost associated with subset M_j . The set multi-cover problem has many applications. In sensor network deployment, a target is often required to be monitored or detected simultaneously by more than one sensor. For example, triangulation-based positioning systems require at least three sensors (i.e., $k_i = 3$ for all $i \in M$) to monitor an object [71]. In an intruder detection and classification system, multiple sensors need to detect an event in order to differentiate between different objects (e.g., a person or a vehicle) and to estimate the speed and the direction of an object [68]. In some situations, multiple sensor coverage is used as redundancy to provide fault

tolerance and ensure system reliability [56]. In ambulance coverage problems, two or more ambulances within a certain range are assigned to each patient for backup coverage [43, 69, 58]. Other applications of the set multi-cover model can be found in client/server protocols [7], wireless sensor networks [1, 91], surveillance system [77], and so forth. The set multi-cover problem is NP-hard. In recent years, the approximability of the set multi-cover problem in restricted settings has been studied and various results presented [7, 40, 21, 73, 112, 20, 35].

We examine stochastic counterparts of set multi-cover problems arising from uncertainties in the input data, e.g., an element i is covered by a subset M_j with some certain probability. For example, the sensing capability of a sensor is not deterministic and the detection of a target by a sensor is random. Various probabilistic models that are used to model the random sensing ability have been proposed, e.g., expressing the probability of a target being detected by a sensor as a function of Euclidean distance between the target and the sensor [68]. In the ambulance coverage problem, each ambulance has the same probability q , called the busy fraction, of being unavailable to answer a call. The busy fraction can be estimated by dividing the total estimated duration of calls for all demand points by the total number of available ambulances.[27].

In our work, we employ a probabilistically-constrained model to address uncertainties. A *Probabilistic Set k -Cover Problem* is to seek a collection of subsets with a minimal cost whose union covers each element at least k times with a probability of at least $1 - \epsilon$, i.e.,

$$\min\left\{\sum_{j \in N} c_j x_j : x \in X \cap Q\right\},$$

where Q is some set defined by deterministic constraints, e.g., budget constraints. X is the set defined by the probabilistic constraint:

$$X := \{x \in \{0, 1\}^n : \mathbb{P}\{\sum_{j=1}^n \tilde{a}_{ij}x_j \geq k\} \geq 1 - \epsilon_i \quad \forall i \in M\}. \quad (74)$$

where \tilde{a}_{ij} is a Bernoulli random variable indicating whether element $i \in M$ is incident to set $j \in N$, k is a positive integer, and $k \leq n$. The parameter $\epsilon_i \in (0, 1)$ is a prescribed risk level for element i . The probabilistic constraint in (74) requires that element $i \in M$ be covered by a collection of subsets for at least k times with a probability of at least $1 - \epsilon_i$. We call set X the *Probabilistic k -Cover Set*.

Research related to this work can be found in the subject of probabilistic set cover problems, i.e., $k = 1$. A version of the probabilistic set cover problem in which the uncertainties only reside in the right-hand vector of the cover inequalities was studied in [18] and [107] and the concept of p -efficient points was employed to build MIP reformulations. In location problems, e.g., the reserve site location [65] and the ambulance location [101], which bear relation to the probabilistic set covering problem, an independence assumption across all columns was made so that deterministic models could be built utilizing this simplified probabilistic structure. The authors in [61] studied a stochastic covering problem in which a feasible covering solution can be built in multiple stages with an adaptive policy or a non-adaptive policy and compared the results of the two policies. Most of the existing research on probabilistic set cover problem with random technology matrix assumes independence across random data. The independence assumption makes it significantly easier to solve a stochastic optimization problem, however, it is often unrealistic. The authors in [3] proposed the concept of *Price Of Correlations* to quantify loss incurred by ignorance of the correlations and pointed out that the loss could be particularly large for some cost functions. The inaccuracy caused by the independence assumption was also acknowledged in probabilistic set cover problems, and a distributionally robust model that incorporated pair-wise correlations was proposed in [5]. However, none of the existing approaches can be extended to general probabilistic set k -cover problems.

In this chapter, we tackle the probabilistic set cover problem with the most general setting, i.e., the coefficients on the left-hand sides are random, data can be correlated, and k can be any integer between 1 and n . Our goal is to develop inner and outer deterministic mixed integer linear programming (MILP) approximations for the probabilistic set defined in (74). Unlike scenario-based approaches for solving probabilistically constrained problems such as the sample average approximation [81], our approach is scenario-free, and the reformulation and bounds obtained are deterministic, requiring no posterior validation.

The remainder of this chapter is organized as follows. In Section 4.2, we first introduce the *ambiguous k -cover set*, which is a “distributionally robust” model for probabilistic k -cover problems. We present a deterministic MILP reformulation of the ambiguous k -cover set based on the so-called *boolean problem* which produces bounds for the probability of k -coverage. However, this reformulation is of exponential size. In order to have a tractable formulation, we aggregate the constraints and variables in the exponential-size formulation to obtain polynomial-size approximations of the original set. In Section 4.3, we build polynomial-size approximations of the ambiguous k -cover set using a LP-based probability bound which can be obtained by aggregating the boolean problem. Then we linearize the approximations and formulate them as mixed integer linear programs. To obtain tighter approximations, in Section 4.4 we improve the bounds for the probability of k -coverage by exerting less aggregation. We also derive extra constraints to strengthen the bounds and numerical results show that the quality of bounds is significantly improved in the strengthened model. In Section 4.5, we demonstrate the usefulness of modeling technique developed for probabilistic k -cover problems by formulating a number of problems that have up till now only been studied under data independence assumption and we also introduce a new applications. Section 4.6 concludes this chapter with possible future research directions.

4.2 Ambiguous Probabilistic k -Cover Set

4.2.1 Definition

For simplicity, we focus on only one row of constraint in (74). The results for the single constraint case can be applied to each row of (74) individually. Let $(\tilde{a}_1, \dots, \tilde{a}_n)$ be a row vector in set (74), whose components are Bernoulli random variables. Fully characterizing the probability of $\tilde{a}_1, \dots, \tilde{a}_n$ requires exponential-size information, i.e., including marginal distributions and all the joint distributions up to order n . Therefore, when n gets large, it is impossible to encode and utilize all the information. Furthermore, the complete distribution function for $(\tilde{a}_1, \dots, \tilde{a}_n)$ is often not available in practice, unless the Bernoulli random numbers \tilde{a}_j are independent from each other. The available information is normally the marginal distributions, pair-wise joint distributions, and sometimes triple-wise joint distributions. With incomplete distribution information, the underlying true probability space and distribution function that reveal the given information cannot be uniquely identified. Therefore, the probabilistic constraint in (74) becomes ambiguous: it is not clear which specific distribution function we refer to when evaluating the probability. Then the uncertainty becomes two-fold: the distribution of $(\tilde{a}_1, \dots, \tilde{a}_n)$ is uncertain; and the values of $(\tilde{a}_1, \dots, \tilde{a}_n)$ are also uncertain. To clarify the ambiguity, we take a robust point of view and use the concept of ambiguous chance constraints [52] to restate the set defined in (74) as follows:

$$X_U = \left\{ x \in \{0, 1\}^n : \mathbb{P}^\zeta \left\{ \sum_{j=1}^n \tilde{a}_j x_j \geq k \right\} \geq 1 - \epsilon \ \forall \zeta \in \mathcal{P}^m \right\},$$

where \mathbb{P}^ζ evaluates the probability that the k -coverage is achieved with regard to the probability distribution ζ ; \mathcal{P}^m represents the set of all possible n -variate Bernoulli distributions ζ with specified marginal distributions and joint distributions up to

order m . We call X_U the *Ambiguous Probabilistic k -Cover Set* or *Ambiguous k -Cover Set*. An optimization problem over an ambiguous k -cover set is called an *Ambiguous k -Cover Problem*. The definition of X_U suggests that the ambiguous k -cover set is a “distributionally robust” variant of the probabilistically constrained sets in the sense that, for a solution x to be admissible, the probabilistic constraint in X_U must be satisfied even by the “worst” distribution ζ that exhibits the revealed information. Clearly, the ambiguous k -cover set is a restriction of the probabilistic k -cover set. Therefore, the ambiguous k -cover problem provides an upper bound for the corresponding probabilistic k -cover problem. It is not hard to see that X_U can be equivalently written as the following set

$$X_U = \left\{ x \in \{0, 1\}^n : \inf_{\zeta \in \mathcal{P}^m} \{ \mathbb{P}^\zeta \left(\sum_{j=1}^n \tilde{a}_j x_j \geq k \right) \} \geq 1 - \epsilon \right\}. \quad (75)$$

A natural counterpart of the ambiguous k -cover set is obtained by substituting the \inf in (75) by \sup , i.e.,

$$X_L = \left\{ x \in \{0, 1\}^n : \sup_{\zeta \in \mathcal{P}^m} \{ \mathbb{P}^\zeta \left(\sum_{j=1}^n \tilde{a}_j x_j \geq k \right) \} \geq 1 - \epsilon \right\}, \quad (76)$$

which resembles an optimistic point of view and is clearly a relaxation for the probabilistic k -cover set and provides a lower bound for the probabilistic k -cover problem. We remark on the definitions in (75) and (76) that given a probabilistic k -cover problem with incomplete distribution information, X_U and X_L provide the best possible inner and outer approximations, respectively. We will focus our attention on inner approximations but also discuss the outer approximations in parallel. In the next section, we develop exact deterministic MILP reformulations for the sets X_U and X_L .

4.2.2 Deterministic Reformulation

The key issue in reformulating X_U and X_L is the lower and upper bounds on the probability of k -coverage given partial information. We calculate the probability bounds using the *Boolean model* [25].

Let $\{\tilde{a}_j : j = 1, \dots, n\}$ be the set of Bernoulli random variables whose probability distributions are given by $\mathbb{P}\{\bigcap_{j \in C} \tilde{a}_j = 1\} = p_C$ for all $C \subseteq N$ ($|C| \leq m$ if only moments up to level m are given). We define another set of events as follows: $A_j := \{\tilde{a}_j x_j = 1\}$ and $\bar{A}_j := \{\tilde{a}_j x_j = 0\}$. Clearly, $\{A_j : j = 1, \dots, n\}$ is also a set of Bernoulli events parametrized by x_j and the probability distributions are functions of x_j : $p_C(x) := \mathbb{P}(\bigcap_{j \in C} A_j) = p_C \prod_{j \in C} x_j$ for any $C \subseteq N$. We denote $\mathbb{P}((\bigcap_{j \in C} A_j) \cap (\bigcap_{j \notin C} \bar{A}_j))$ as v_C for all $C \subseteq N$. Note that v_C s represent the probabilities of all the outcomes in the sample space which are mutually exclusive. Then the probability at least k out of n events occurring, given a vector x , can be expressed as $\sum_{C \subseteq N, |C| \geq k} v_C$ and its lower bound (or upper bound) can be obtained by the Boolean linear programming model [25, 66] defined as follows

$$\begin{aligned}
& \min(\max) \quad \sum_{C \subseteq N, |C| \geq k} v_C \\
& \text{s.t.} \quad \sum_{C \subseteq N} v_C = 1 \\
& \quad \sum_{S \subseteq C} v_C = p_S(x), \quad \forall S \subseteq N, |S| \leq m \\
& \quad v_C \geq 0 \quad \forall C \subseteq N.
\end{aligned} \tag{77}$$

As (77) consists of variables representing the probabilities of every possible outcome in the sample space, any bound based on p_S $|S| \leq m$ can be restated as the objective function of the Boolean problem. For example, the probability of at least k out of n events can be recovered as the objective function in (77). Therefore, the bounds on the probability of certain events produced by the Boolean problem are the best possible bounds given p_S . We use the lower bound produced by the Boolean problem to construct a deterministic reformulation for the ambiguous probabilistic k -cover set.

Proposition 15. *Let X_U be the ambiguous k -cover set defined in (75) and \mathcal{P}^m be the family of distributions with moments up to degree m available, i.e., for any $\zeta \in \mathcal{P}^m$,*

we have $\mathbb{P}(\bigcap_{j \in S} \tilde{a}_j = 1) = p_S$ for all $S \subseteq N$ and $|S| \leq m$. Let

$$X_B = \{x \in \{0, 1\}^n : \min\{\sum_{\substack{C \subseteq N \\ |C| \geq k}} v_C : \sum_{C \subseteq N} v_C = 1, \sum_{S \subseteq C} v_C = p_S(x), \forall S \subseteq N, |S| \leq m, v_C \geq 0\} \geq 1 - \epsilon\}, \quad (78)$$

where $p_S(x) = p_S \prod_{j \in S} x_j$. Then

$$X_U = X_B$$

Proof. $X_B \subseteq X_U$: Let $x^* \in X_B$, then for any $\zeta \in \mathcal{P}^m$ we have

$$\begin{aligned} \mathbb{P}^\zeta\{\sum_{j=1}^n \tilde{a}_j x_j^* \geq k\} &= \sum_{\substack{C \subseteq N \\ |C| \geq k}} v_C : \sum_{C \subseteq N} v_C = 1, \sum_{S \subseteq C} v_C = p_S(x^*) \quad \forall S \subseteq N, |S| \leq m \\ &\geq \min\{\sum_{\substack{C \subseteq N \\ |C| \geq k}} v_C : \sum_{C \subseteq N} v_C = 1, \sum_{S \subseteq C} v_C = p_S(x^*) \quad \forall S \subseteq N, |S| \leq m\} \\ &\geq 1 - \epsilon. \end{aligned}$$

Therefore, $x^* \in X_U$.

$X_U \subseteq X_B$: Let any $x^* \notin X_B$, we show $x^* \notin X_U$ either. When $x^* \notin X_B$, we have

$$z^* = \min\{\sum_{\substack{C \subseteq N \\ |C| \geq k}} v_C : \sum_{C \subseteq N} v_C = 1, \sum_{S \subseteq C} v_C = p_S(x^*) \quad \forall S \subseteq N, |S| \leq m\} < 1 - \epsilon, \quad (79)$$

where p_S are given moments up to level m .

We need to show that there is a distribution $\hat{\zeta} \in \mathcal{P}^m$ such that $\mathbb{P}^{\hat{\zeta}}(\sum_{j=1}^n \tilde{a}_j x_j^* \geq k) < 1 - \epsilon$. Given x^* , let v_C^* be an optimal solution of the minimization problem (79) inside the definition of X_B . We construct a Bernoulli distribution $\hat{\zeta}$ for $\{\tilde{a}_j, j \in N\}$ by recovering from v_C^* :

$$\hat{p}_S = \sum_{C: C \subseteq N, S \subseteq C} v_C^* \quad \forall S \subseteq N$$

Since v_C^* s are mutually exclusive and $\sum_{C \subseteq N} v_C^* = 1$, v_C^* s are valid probabilities.

Therefore, \hat{p}_S are also valid probabilities. From the construction, we know that $\hat{p}_S^* = p_S$ for all $S \subseteq N$ and $|S| \leq m$. Therefore, $\hat{\zeta} \in \mathcal{P}$.

Given $\hat{\zeta}$ and x^* , $\mathbb{P}^{\hat{\zeta}}(\sum_{j=1}^n \tilde{a}_j x_j^* \geq k) = \sum_{C \subseteq N, |C| \geq m} \bar{v}_C$, where \bar{v}_C is the unique solution to the following complete Boolean problem:

$$\begin{aligned} \sum_{C \subseteq N} v_C &= 1 \\ \sum_{S \subseteq C} v_C &= \hat{p}_S(x^*), \quad \forall S \subseteq N, \\ v_C &\geq 0 \quad \forall C \subseteq N \end{aligned} \tag{80}$$

Since v_C^* is a feasible solution to the equation system above, $\bar{v}_C = v_C^*$ (to see this, notice that the above equation system consists of 2^n equations and 2^n decision variables. By permuting the rows of the equation system, we can rearrange the left-hand side matrix as an upper triangular matrix. Hence, the rank of the equation system is 2^n , and hence a unique solution is determined if a valid probability distribution is given on the right-hand side). Then, $\mathbb{P}^{\zeta^*}(\sum_{j=1}^n \tilde{a}_j x_j^* \geq k) = \sum_{C \subseteq N, |C| \geq m} \bar{v}_C = \sum_{C \subseteq N, |C| \geq m} v_C^* = z^* < 1 - \epsilon$ and hence $x^* \notin X_U$.

We conclude that $X_U = X_B$.

□

A deterministic reformulation for X_L can be obtained by simply changing the minimization operator in (78) to a maximization operator.

Although Proposition 15 gives a deterministic reformulation of the ambiguous k -cover set, X_B consists of exponential number of auxiliary variables v_C regardless of m . One might think that, since there are only polynomial number of constraints in the boolean model, we may obtain a tractable description for X_B by projecting the auxiliary variables out. Unfortunately, the resulting projection has exponential number of facets. Therefore, the question now becomes whether we can approximate the ambiguous k -cover set with a tractable model with reasonable accuracy.

We utilize a series of linear program based bound results on the probabilities of certain events [98, 99, 26, 100] to develop polynomial-size models that approximate

the ambiguous k -cover set X_B in the rest of this chapter. The technique that we use is based on the following observation: in a linear program, if several variables share the same coefficients on each row and the objective function, we can aggregate these variables into a single variable, and hence reduce the number of variables. We will apply this technique to the Boolean problem, by aggregating rows of equations in (77) to create columns that share the same coefficients at each row and then aggregating these columns. Notice that LP optimal values of the aggregated linear programs will be valid lower bounds on the probability of k -coverage, since the validity of the aggregated formulation is implied by the original Boolean LP. Sections 4.3 and 4.4 discuss the models obtained by different levels of aggregations.

4.3 Fully Aggregated Model (FAM)

The first aggregation is to add up all rows with the same level of moments, and then we arrive at the following equations:

$$\sum_{t=1}^n \binom{t}{i} \left(\sum_{C \subseteq N: |C|=t} v_C \right) = \sum_{S \subseteq N: |S|=i} p_S(x) \quad i = 1, \dots, m. \quad (81)$$

Then we substitute $\sum_{C \subseteq N: |C|=t} v_C$ with a new variable v_t ($v_t \geq 0$) and obtain the following linear system:

$$\begin{aligned} v_0 + v_1 + v_2 + v_3 + v_4 + \dots + v_n &= 1 \\ v_1 + 2v_2 + 3v_3 + 4v_4 + \dots + nv_n &= s_1(x) \\ v_2 + \binom{3}{2}v_3 + \binom{4}{2}v_4 + \dots + \binom{n}{2}v_n &= s_2(x) \\ v_3 + \binom{4}{3}v_4 + \dots + \binom{n}{3}v_n &= s_3(x) \\ &\vdots \\ v_m + \dots + \binom{n}{m}v_n &= s_m(x) \end{aligned} \quad (82)$$

where $s_i(x) = \sum_{S \subseteq N: |S|=i} p_S(x)$. The new variable v_i has a physical meaning: the probability that exact i events occur. Let us denote (82) as $T_m v = S_m(x)$, where $v = (v_0, v_1, \dots, v_n) \in \mathbb{R}_+^{n+1}$, $T_n \in \mathbb{R}^{(m+1) \times (n+1)}$ is the left-hand side coefficients, and

$S_m(x) = (s_0, s_1, \dots, s_m) \in \mathbb{R}_+^{m+1}$ is the $(m+1)$ -dimensional vector on the right-hand side. For convenience, we define $\binom{n_1}{n_2} = 0$ when $n_1 < n_2$.

The Boolean model X_U , after aggregation and variable substitution, becomes the following linear program:

$$\min \left\{ \sum_{i=k}^n v_i : T_m v = S_m(x), v \in \mathbb{R}_+^{n+1} \right\}, \quad (83)$$

We can also aggregate X_L in the same way and obtain the following maximization problem

$$\max \left\{ \sum_{i=k}^n v_i : T_m v = S_m(x), v \in \mathbb{R}_+^{n+1} \right\} \quad (84)$$

We call as *Fully-Aggregated Model* or *FAM*. For a fixed vector x , the linear programs in (83) and (84) were first developed by Prékopa and other researchers in a series of papers [98, 99, 26], using the concept of *binomial moments*, to calculate bounds for the probabilities of certain events. Here we borrow the name and call $s_m(x)$ the m -th binomial moments.

Let us denote the optimal objective value of the minimization problem in (83) as $f^L(x)$ and the optimal objective value of the minimization problem in (77) as $z^B(x)$. Clearly, $f^L(x)$ is a lower bound for $z^B(x)$ for any vector x . Let us denote the optimal objective value of (84) as $f^U(x)$ and the optimal objective value of the maximization problem in (77) as $z^L(x)$. Clearly, $f^U(x)$ is an upper bound for $z^U(x)$ for a given vector x . We observe that

Observation 1. $f^L(x) = f^U(x) = z^B(x)$ when $m = n$.

Proof. When $m = n$, the Boolean problem (77) has 2^n equations which can be rearranged into an upper-triangular matrix. Therefore, it has a unique solution and the solution is feasible since the right-hand sides are valid probability distributions. The two fully aggregated linear programs (83) and (84) have n equations and the left-hand sides can also be rearranged as an upper-triangular matrix, therefore, both

LPs have the same unique solution. Let v_C^* be the unique solution to (77), then $v_i^* = \sum_{C \subseteq N: |C|=i} v_C^*$ $i = 0, 1, \dots, n$ is also a feasible solution to both (83) and (84), which must be the optimal one. Therefore, the objective function values of the three optimal solutions are equal. \square

Observation 1 suggests that when provided with sufficient information, bounds produced by the FAM can be tight.

4.3.1 Inner and Outer Approximations

With the lower and upper bounds for the probability of k -coverage introduced in the previous section, we develop inner and outer approximations for set X . First, we define the following two sets:

$$X_{RF} := \{x \in \{0, 1\}^n : \max\{\sum_{i=k}^n v_i : T_m v = S_m(x), v \geq 0\} \geq 1 - \epsilon\} \quad (85)$$

and

$$X_{SF} := \{x \in \{0, 1\}^n : \min\{\sum_{i=k}^n v_i : T_m v = S_m(x), v \geq 0\} \geq 1 - \epsilon\}. \quad (86)$$

Proposition 16. X_{SF} is a restriction of X and X_{RF} is a relaxation of X .

Proof. X_{SF} is a restriction of X : It is sufficient to show that X_{SF} is a restriction of X_B : Let $\hat{x} \in X_{SF}$, then $f^L(\hat{x}) \geq 1 - \epsilon$, where $f^L(\hat{x})$ is the optimal value of the minimization problem (83). Since $f^L(\hat{x})$ is a lower bound for $z^B(\hat{x})$, then $z^B(\hat{x}) \geq 1 - \epsilon$. Therefore, $\hat{x} \in X_B$.

X_{RF} is a relaxation of X : It is sufficient to show that X_{RF} is a relaxation of X_L : Let $\hat{x} \in X_L$, then $z^L(\hat{x}) \geq 1 - \epsilon$, where $z^L(\hat{x})$ is the optimal solution of the maximization problem in (77). Since $f^U(\hat{x}) \geq z^L(\hat{x}) \geq 1 - \epsilon$, $\hat{x} \in X_{RF}$. \square

4.3.2 MIP Formulations

In order to build MIP formulations, we first remove the optimization problems in the definition of the approximations.

Proposition 17. $X_{RF} = \{x \in \{0, 1\}^n : \exists v \in \mathbb{R}_+^{n+1} \text{ s.t. } \sum_{i=k}^n v_i \geq 1 - \epsilon, T_m v = S_m(x)\}.$

Proof. \subseteq : Let $\hat{x} \in X_{RF}$, and v^* be the corresponding optimal solution of the maximization problem in (85), i.e., $\max\{\sum_{i=k}^n v_i : T_m v = S_m(\hat{x}), v \geq 0\}$. Then there is a $v = v^*$ such that $\sum_{i=k}^n v_i \geq 1 - \epsilon$ and $T_m v = S_m(\hat{x})$. Therefore, $\hat{x} \in X_{RF}$.

\supseteq : Let \hat{x} be a vector in the set defined on the right-hand side of the equation in the proposition, then $\exists \hat{v}$ such that $\sum_{i=k}^n \hat{v}_i \geq 1 - \epsilon$ and $T_m \hat{v} = S_m(\hat{x})$. Therefore, the maximization problem in (85), $\max\{\sum_{i=k}^n v_i : T_m v = S_m(\hat{x}), v \geq 0\}$, is feasible and (\hat{x}, \hat{v}) is a feasible solution. Let v^* be the optimal solution to the maximization problem in (85), then $\sum_{i=k}^n v_i^* \geq \sum_{i=k}^n \hat{v}_i \geq 1 - \epsilon$. Hence, $\hat{x} \in X_{RF}$. \square

As for X_{SF} , let $\pi \in \mathbb{R}^{m+1}$ be the dual variables associated with the equations in the minimization problem in (86) and e_k be the $(n+1)$ -dimensional objective vector with 0 on the first k positions and 1 on the rest.

Proposition 18. $X_{SF} = \{x \in \{0, 1\}^n : \exists \pi \text{ s.t. } \pi^\top S_m(x) \geq 1 - \epsilon, \pi^\top T_m \leq e_k\}.$

Proof. Since the minimization problem in (86) is feasible and bounded below, from strong duality theorem, X_{SF} defined in (86) is equivalent to the following set

$$\{x \in \{0, 1\}^n : \max\{\pi^\top S_m(x) : \pi^\top T_m \leq e_k, \pi \in \mathbb{R}^{m+1}\} \geq 1 - \epsilon\},$$

where $\pi \in \mathbb{R}^{m+1}$ are the dual variables corresponding to the equations of the minimization problem in (86). Then the result follows from identical arguments used in Proposition 17, \square

With a slight abuse of notations, we omit the “ \exists ” statement in the definition of X_{RF} and X_{SF} :

$$X_{RF} = \{x \in \{0, 1\}^n : T_m v = S_m(x), \sum_{i=k}^n v_i \geq 1 - \epsilon, v \geq 0\},$$

and

$$X_{SF} = \{x \in \{0, 1\}^n : \pi^\top T_m \leq e_k, \pi^\top S_m(x) \geq 1 - \epsilon, \pi \in \mathbb{R}^{m+1}\}.$$

Since $S_m(x)$ consists of m multi-linear functions of binary variable x , we linearize them using McCormick relaxation technique [88]. This yields a mixed integer linear programming formulation and enables us to optimize over X_{RF} and X_{SF} using the state-of-art mixed-integer programming technologies.

In X_{RF} , the nonlinearity arises from the product of binary variables in $S_m(x)$. Let a product of binary variables in the t -th moment to be $\Pi_{j \in C} x_j$, where $C \in \mathcal{I}^t := \{I \subseteq N : |I| = t\}$. We use an auxiliary variable y_C^t to represent this product. Since McCormick relaxations are exact on the variable bounds, and the x variables in our problem are 0-1, the following McCormick relaxations guarantee that y_C^t equals $\Pi_{j \in C} x_j$ on 0 and 1 [60]:

$$\begin{aligned} y_C^t &\leq x_j \quad \forall j \in C \\ y_C^t &\geq 1 + (\sum_{j \in C} x_j - t) \\ 0 &\leq y_C^t \leq 1. \end{aligned}$$

We apply the McCormick relaxation technique above to each multilinear term in $S_m(x)$ and obtain the following mixed-integer program formulation for X_{RF} :

Proposition 19.

$$\begin{aligned} \sum_{i=0}^n v_i &= 1 \\ \sum_{i=t}^n \binom{i}{t} v_i &= \sum_{C \in \mathcal{I}^t} p_C^t y_C^t \quad t = 1, \dots, m \\ y_C^t &\leq x_j \quad \forall j \in C, \forall C \in \mathcal{I}^t, t = 2, \dots, m \\ X_{RF} &= \{x \in \{0, 1\}^n, \quad y_C^t \geq 1 + (\sum_{j \in C} x_j - t) \quad \forall C \in \mathcal{I}^t, t = 2, \dots, m \quad \}. \quad (87) \\ \sum_{i=k}^n v_i &\geq 1 - \epsilon \\ 0 &\leq y_C^t \leq 1 \quad \forall C \in \mathcal{I}^t, \quad t = 1, \dots, m \\ v_i &\geq 0 \quad i = 0, 1, \dots, n \end{aligned}$$

The linearization for X_{SF} is different from that for X_{RF} due to the continuous variables π_t . Assuming $\pi_t \in [l_t, u_t]$, we define $M_t^+ = \max\{0, u_t\}$ and $M_t^- = \min\{0, l_t\}$, the the nonlinear term $\pi_t \prod_{j \in C} x_j$ can be linearized using the following formulation:

$$\begin{aligned}
y_C^t &\leq M_t^+ x_j \quad \forall j \in C \\
y_C^t &\geq M_t^- x_j \quad \forall j \in C \\
y_C^t &\leq \pi_t - M_t^-(t - \sum_{j \in C} x_j) \\
y_C^t &\geq \pi_t - M_t^+(t - \sum_{j \in C} x_j).
\end{aligned} \tag{88}$$

Then the mixed-integer program reformulation for X_{SF} can be obtained by replacing each nonlinear term by the linearization in (88):

Proposition 20.

$$\begin{aligned}
\pi_0 + \sum_{t=1}^m \binom{i}{t} \pi_t &\leq c_k^i \quad i = 1, \dots, n \\
\pi_0 + \sum_{t=1}^m \sum_{C \in \mathcal{I}^t} p_C^t y_C^t &\geq 1 - \epsilon \\
y_C^t &\leq M_t^+ x_j \quad \forall j \in C, \forall C \in \mathcal{I}^t, t = 1, \dots, m \\
X_{SF} = \{x \in \{0, 1\}^n, \quad &y_C^t \geq M_t^- x_j \quad \forall j \in C, \forall C \in \mathcal{I}^t, t = 1, \dots, m \\
&y_C^t \leq \pi_t - M_t^-(t - \sum_{j \in C} x_j) \quad \forall C \in \mathcal{I}^t, t = 1, \dots, m \\
&y_C^t \geq \pi_t - M_t^+(t - \sum_{j \in C} x_j) \quad \forall C \in \mathcal{I}^t, t = 1, \dots, m \\
&\pi_0 \leq 0; \quad \pi_t \text{ free } t = 1, \dots, n; \quad y_C^t \geq 0 \quad \forall C \in \mathcal{I}^t, t = 1, \dots, m
\end{aligned} \tag{89}$$

The dual variables π_t are unbounded. However, since we know that the primal problem in (86) is feasible and bounded below for any 0-1 vector x , the dual problem must also be feasible and bounded on the directions of dual objectives. Therefore, for any objective function $S_m(x)$ of the dual problem, there must be an extreme point which is optimal. Thus, we restrict our attention to the extreme points and derive bounds for the dual variables π . Therefore, instead of supplying M with arbitrarily large values, we derive tight bounds for the dual variables in the following proposition, which help tighten LP relaxations subsequently. We present the results for the case

when $m = 2$. In practice, the available distribution information is often the marginal distributions and pair-wise joint distributions, i.e., $m = 2$.

Proposition 21. *With m fixed to 2,*

1. *If $k = 1$, then*

$$\pi_0 = 0, \pi_1 \in [\frac{2}{n}, 1], \text{ and } \pi_2 \in [-1, -\frac{2}{n(n-1)}]. \quad (90)$$

2. *If $k = 2$, then*

$$\pi_0 \in [-2, 0], \pi_1 \in [0, 2], \text{ and } \pi_2 \in [-1, \frac{2}{n(n-1)}]. \quad (91)$$

3. *If $k \geq 3$, then*

$$\pi_0 \in [-\frac{(k-1)(k+2)}{2}, 0], \pi_1 \in [-\frac{k-2}{n(n-k+1)}, k], \text{ and } \pi_2 \in [-1, \frac{2}{n(n-k+1)}]. \quad (92)$$

Proof. When $m = 2$, the dual polyhedron $\pi^\top T_m \leq e_k$ can be written as follows

$$\begin{aligned} \pi_0 &\leq 0 \\ \pi_0 + i\pi_1 + \binom{i}{2}\pi_2 &\leq 0 \quad i = 1, \dots, k-1 \\ \pi_0 + i\pi_1 + \binom{i}{2}\pi_2 &\leq 1 \quad i = k, \dots, n \\ \pi_0, \pi_1, \pi_2 &\text{ free,} \end{aligned}$$

where the rows are numbered from 0 to n . We examine all possible extreme points and determine bounds for the dual variables. Since there is no hidden equality, the dual polyhedron is full-dimensional. Therefore, there are three linearly independent constraints binding at each extreme point. By the results of dual feasible bases in Theorem 7.4 in [26], the extreme points can be formed by only three types of choices of rows. Let the set of rows that form an extreme point be I ($|I| = 3$), the three types of choices are

1. $I \in \{0, 1, \dots, k-1\}$ if $k \geq 3$

Since all constraints in I are linearly independent, $\pi_0 = \pi_1 = \pi_2 = 0$.

2. $I = \{0, k-1, n\}$ if $k \geq 2$

$$\pi_0 = 0, \pi_1 = -\frac{k-2}{n(n-k+1)}, \text{ and } \pi_2 = \frac{2}{n(n-k+1)}.$$

3. $I = \{k-1, i, i+1\}$ if $1 \leq k \leq n-1, i \in \{k, \dots, n-1\}$

Solving the equation system, we have

$$\pi_0 = -\frac{(k-1)(2i-k+2)}{(i-k+1)(i-k+2)}, \pi_1 = \frac{2i}{(i-k+1)(i-k+2)}, \pi_2 = -\frac{2}{(i-k+1)(i-k+2)}.$$

Clearly, π_0 is a monotonically increasing function of i and $\lim_{i \rightarrow +\infty} \pi_0 = 0$. Therefore $0 \geq \pi_0 \geq \pi_0(k) = -\frac{(k-1)(k+2)}{2}$. π_1 is a monotonically decreasing function of i and $\lim_{i \rightarrow +\infty} \pi_1 = 0$. Therefore, $0 \leq \pi_1 \leq \pi_1(k) = k$. Similarly, π_2 is a monotonically increasing function of i and $\lim_{i \rightarrow +\infty} \pi_2 = 0$. Therefore, $0 \geq \pi_2 \geq \pi_2(k) = -1$.

We take the maximal and minimal possible values that each variable can take, arriving at the conclusion in the proposition. □

4.3.3 Numerical Examples

To gain some empirical experience on the quality and convergence properties of the bounds with regard to m , we conduct computational experiments on randomly generated instances.

The instances are probabilistic sensor multi-coverage problems in which a set of sensors can be deployed to n candidate locations to monitor targets. A target is detected by a sensor with a certain probability affected by the sensing ability and the distance between the sensor and the target [68]. And whether the target is detected by a sensor at location j is modeled as a Bernoulli random number \tilde{a}_j . A target is considered as “covered” if it is detected by at least three different sensors [71], and

we require that the probability of being covered for each target is at least $1 - \epsilon$. The decision is to install a minimal number of sensors on a set of candidate locations, while maintaining the coverage reliability for each target. This problem can be modeled by the probabilistic k -cover model as follows

$$\begin{aligned}
& \min \sum_{j=1}^n x_j \\
& \text{s.t. } \mathbb{P}\left(\sum_{j=1}^n \tilde{a}_j x_j \geq 3\right) \geq 1 - \epsilon \\
& x_j \in \{0, 1\} \quad j = 1, \dots, n.
\end{aligned} \tag{93}$$

Notice that, to focus on the basic probabilistic k -cover set, we only consider the case with a single target. Since we assume that distribution function is incomplete, we formulate (93) as an ambiguous k -cover problem:

$$\begin{aligned}
& \min \sum_{j=1}^n x_j \\
& \text{s.t. } \inf_{\zeta \in \mathcal{P}^m} \mathbb{P}^\zeta\left(\sum_{j=1}^n \tilde{a}_j x_j \geq 3\right) \geq 1 - \epsilon \\
& x_j \in \{0, 1\} \quad j = 1, \dots, n.
\end{aligned} \tag{94}$$

The reliability level $1 - \epsilon$ is set to 0.8 and $n=10$. The probability $\mathbb{P}(\tilde{a}_j = 1)$ is drawn from a uniform distribution between 0.5 and 0.8. For the convenience of setting up joint distributions, we assume that the detections by different sensors are independent. We randomly generate 20 instances. For each instance, we first use the Boolean model (minimization problem in (77)) based formulations X_B defined in (78) to reformulate (94) as an ambiguous k -cover problem and obtain an upper bound. Notice that X_B produces the best possible upper bounds for the true coverage cost. Then we calculate upper bounds for the coverage cost using FAM based restrictions as in (89). In the meanwhile, we also use the BM (maximization problem in (77)) based MIP

and the FAM based relaxations (87) to calculate lower bounds for the true coverage costs. We calculate these bounds with different levels of available information, i.e., $m = 2, 3, \dots, 10$, and compare their strength. We observe that all the 20 instances exhibit similar features in terms of bounds quality and convergence rates. Therefore, we present four typical instances in the figures in Figure 2.

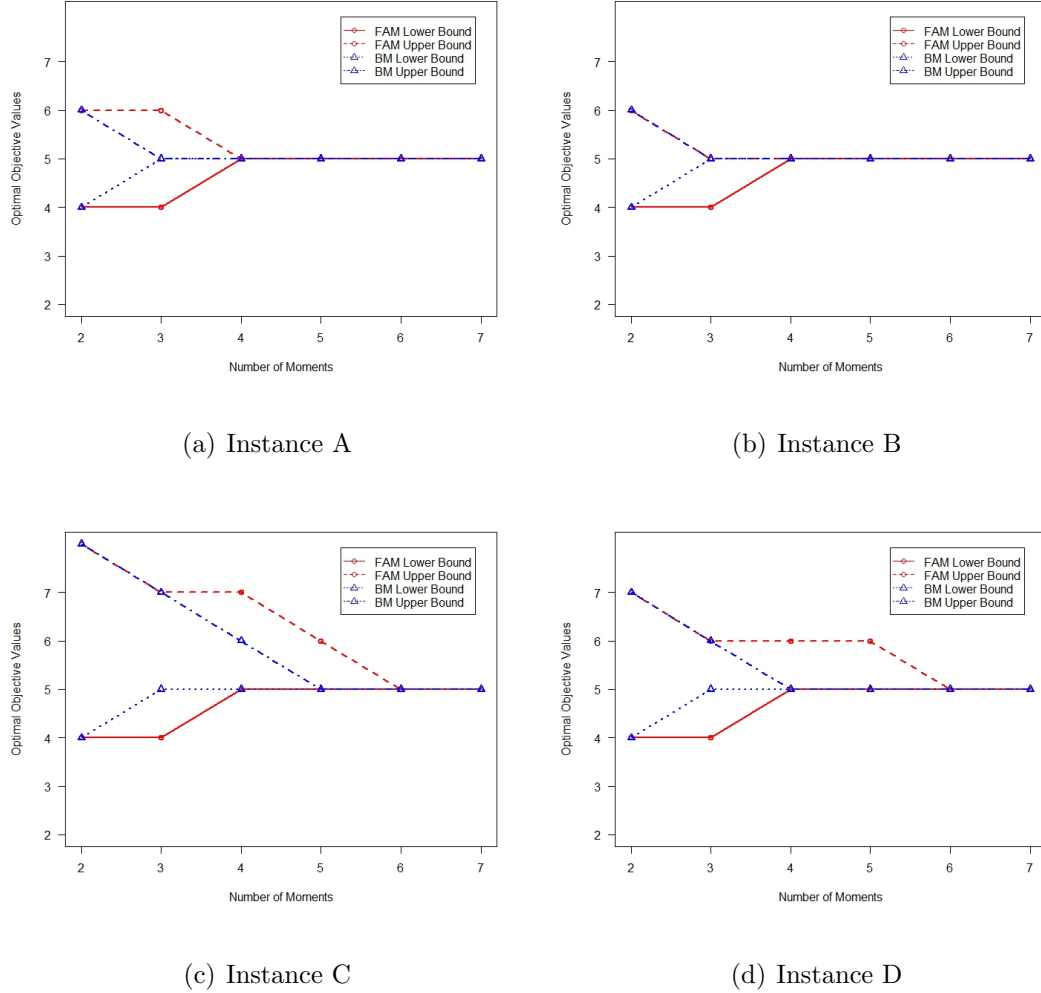


Figure 2: Bounds for 3-Coverage Single Target Instances ($n=10$)

The figures in Figure 2 show that:

1. Given a small amount of information, e.g., $m=2$, the upper bounds produced by

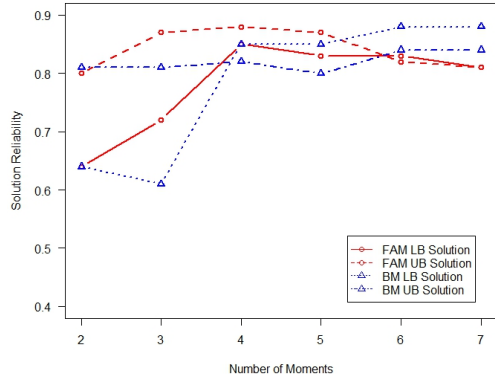
BM are conservative with high costs, in order to maintain feasibility for all possible distributions; when more information is incorporated into the model, i.e., m increases, the number of possible distributions in the family \mathcal{P}^m is reduced, and so are the costs.

2. When m is sufficiently large, lower and upper bounds provided by FAM are equal. We observe that for the majority of the instances, the lower and upper bounds produced by FAM converge when $m = 4$, which indicates that when sufficient information is provided, the bounds produced by FAM can be tight.
3. Given the same amount of information, the bounds yield by BM are at least as strong as those by FAM. However, the recorded solution time for BM formulations is significantly longer.

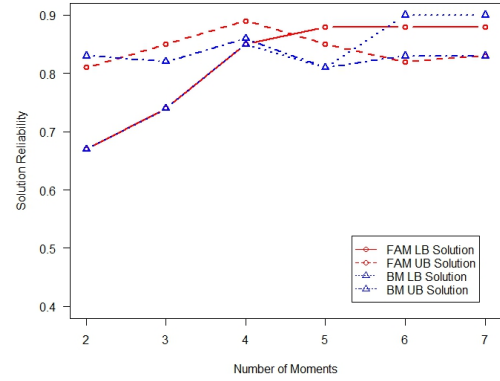
Next, we examine the reliability of solutions produced by the relaxations and restrictions based on FAM and BM. Let $\bar{x} \in \{0, 1\}^n$ be a solution, the reliability r of \bar{x} is defined, with regard to the “distributionally robust” ambiguous k -cover set, as

$$r = \inf_{\zeta \in \mathcal{P}^m} \{\mathbb{P}^\zeta(\sum_{j=1}^n \tilde{a}_j \bar{x}_j) \geq k\},$$

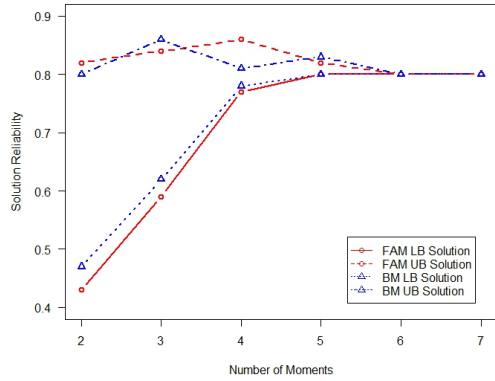
which can be calculated by the Boolean model in (77) when n is not large. If $r \geq 1 - \epsilon$, then \bar{x} is a feasible solution regardless of whichever the real underlying distribution might be. We calculate the reliability of the solutions produced by the four models for the instances in Figure 2 and plot them in Figure 3.



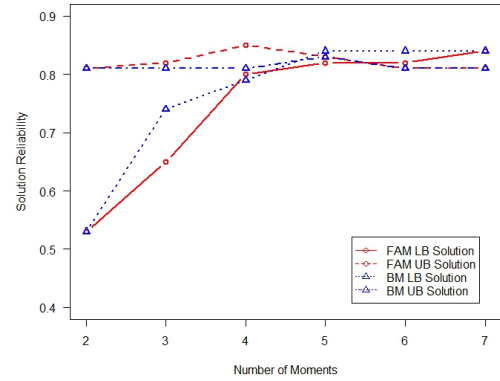
(a) Instance A



(b) Instance B



(c) Instance C



(d) Instance D

Figure 3: Solution Reliability with Single Target Instances

The “BM LB Solution”, “BM UB Solution”, “FAM LB Solution”, and “FAM UB Solution” represent the reliability of the solutions produced by the BM-based relaxations, BM-based restrictions, FAM-based relaxations, and FAM-based restrictions, respectively. We observe that, the solutions produced by BM and FAM-based restrictions are always feasible, as we expected. There is no clear pattern showing that the solutions of one model is more reliable than the other. And a solution’s reliability does not exhibit any obvious sign of relation with its cost. For the lower bound models, i.e., BM-based relaxations and FAM-based relaxations, reliability improves when m increases: when m is larger, it is more likely to obtain a feasible solution. But the

feasibility of a solution produced by a relaxation is not guaranteed.

4.4 *Partially Aggregated Model (PAM)*

We have seen in Section 4.3.3 that, BM-based formulations provide tighter approximations than FAM, by better utilizing the given information. In the full-aggregation process described in (81), when we add up the probabilities at the same level in the Boolean LP (77), we lose the individual probability information, which cannot be recovered in the linear program (82). Since our approximations are built upon the bound results for the probability of k -coverage, the quality of the approximations relies heavily on the quality of the probability bounds. In this section, aiming to provide better bounds than FAM by preserving more information but still maintaining a manageable size, we partially aggregate the Boolean LP. We also derive extra constraints to strengthen this bounding LP.

4.4.1 Partially Aggregated Model

By slightly abusing notation, we drop x in $p_S(x)$ for the current moment. In the partial aggregation scheme, we first duplicate each row with right-hand side $p_S(x)$ in (77) $|S|-1$ times, add up rows with $|S| = i$ and $j \in S$ for each i and j , and then we arrive at

$$\min : \sum_{i=k}^n \sum_{j=1}^n \frac{1}{i} \sum_{C: \substack{|C|=i \\ j \in C}} v_C \quad (95)$$

$$\sum_{i=1}^n \sum_{j=1}^n \frac{1}{i} \sum_{C: \substack{|C|=i \\ j \in C}} v_C = 1 \quad (96)$$

$$\sum_{i=1}^n \sum_{C: \substack{|C|=i \\ j \in C}} v_C = p_j \quad j = 1, \dots, n \quad (97)$$

$$\sum_{i=t}^n \binom{i-1}{t-1} \sum_{C: \substack{|C|=i \\ j \in C}} v_C = \sum_{S: \substack{|S|=t \\ j \in S}} p_S \quad t = 2, \dots, m \quad j = 1, \dots, n \quad (98)$$

$$v_C \geq 0 \quad \forall C \subseteq N. \quad (99)$$

Equations (95)-(99) are the resulting rows by row duplication and aggregation. Notice that the following variables share the same coefficient in each row: v_C s with $|C| = i$ and $j \in C$. Therefore, we can aggregate these variables into a single variable, v_{ij} , which can be interpreted as the probability of i events, including event j , occur, i.e.,

$$v_{ij} = \sum_{C: \substack{|C|=i \\ j \in C}} v_C \quad i = 1, \dots, n \quad j = 1, \dots, n. \quad (100)$$

Equations (100) define links between the old and new variables. To finish the aggregation process, that is, projecting out the variables v_C in the aggregated model and obtaining a model with new variables v_{ij} , we carry out the following two steps:

1. Substitute $\sum_{C: \substack{|C|=i \\ j \in C}} v_C$ in (95)-(99) with v_{ij} ;
2. Project v_C out in (100).

In the first step, equations (95)-(99) with old variables substituted by v_{ij} become

the following linear program:

$$\min : \sum_{i=k}^n \sum_{j=1}^n \frac{1}{i} v_{ij} \quad (101)$$

$$\sum_{j=1}^n \sum_{i=0}^n \frac{1}{i} v_{ij} = 1 \quad (102)$$

$$\sum_{i=t}^n \binom{i}{t} \frac{1}{i} v_{ij} = \frac{1}{t} s_t^j \quad t = 1, \dots, m \quad j = 1, \dots, n \quad (103)$$

$$v \geq 0 \quad i = 1, \dots, n \quad j = 1, \dots, n. \quad (104)$$

where $s_t^j = \sum_{S: |S|=t} p_S$. We call the linear program defined by (101)-(104) the *Partially Aggregated Model* or *PAM*. The partially aggregated model with $k = 1$ first appeared in a linear program developed by Prékopa and Gao in [100] to calculate the lower bound for the probability of a union of events. They derived this model from the perspective of probability using the concept of “disaggregated binomial moments” (s_t^j). Notice that in the case $k = 1$, constraint (102) can be dropped, and then the linear program is separable and a closed-form optimal solution can be obtained. From the aggregation process, it is straightforward that, the model in (95)-(99) also works for more general cases where $k \geq 2$.

For completeness, in the next proposition, we show from the perspective of probability that PAM can be used to calculate the k -coverage probability when $k \geq 2$, extending the results in [100]. We derive only the objective function; the constraints (102)-(104) are the same as those in [100]

Let $\{A_i : j \in N\}$ be a set of events, where N is the index set $\{1, \dots, n\}$. Define random variable $X_j : A_j \rightarrow \{0, 1\}$ as $X_j = 1$ if A_j occurs, and $X_j = 0$, otherwise. Define μ as a random variable that represents the number of events, A_j s, that occur, i.e., $\mu = \sum_j X_j$ and $\mathbb{P}(\mu = i) = v_i$. Let k be a positive integer between 2 and n ; we are interested in the probability $\mathbb{P}(\mu \geq k)$, i.e., the probability that at least k out of n events occur. Clearly, we have $\mathbb{P}(\mu \geq k) = \sum_{i=k}^n v_i$. Then

Proposition 22.

$$\mathbb{P}(\mu \geq k) = \sum_{i=k}^n \sum_{j=1}^n \frac{1}{i} v_{ij},$$

where $v_{ij} = \mathbb{P}(X_j = 1, \mu = i)$.

Proof.

$$\mathbb{P}(\mu \geq k) = \sum_{i=k}^n v_i = \sum_{i=k}^n \mathbb{P}\left(\bigcup_{\substack{S \subseteq N \\ |S|=i}} \left(\bigcap_{\ell \in S} X_\ell = 1, \mu = i\right)\right) = \sum_{i=k}^n \sum_{\substack{S \subseteq N \\ |S|=i}} \mathbb{P}\left(\bigcap_{\ell \in S} X_\ell = 1, \mu = i\right).$$

The last equality holds because for any pair of distinct subsets S_1 and S_2 with $|S_1| = |S_2| = i$, the probability $\mathbb{P}\left(\left(\bigcap_{\ell \in S_1} X_\ell = 1, \mu = i\right) \cap \left(\bigcap_{\ell \in S_2} X_\ell = 1, \mu = i\right)\right) = \mathbb{P}\left(\bigcap_{\ell \in S_1 \cup S_2} X_\ell = 1, \mu = i\right) = 0$ since $|S_1 \cup S_2| > i$. Notice that

$$\begin{aligned} \sum_{\substack{S \subseteq N \\ |S|=i}} \mathbb{P}\left(\bigcap_{\ell \in S} X_\ell = 1, \mu = i\right) &= \frac{1}{i} \sum_{j=1}^n \sum_{\substack{S \subseteq N, |S|=i \\ j \in S}} \mathbb{P}\left(\bigcap_{\ell \in S} X_\ell = 1, \mu = i\right) \\ &= \frac{1}{i} \sum_{j=1}^n \mathbb{P}\left(\bigcup_{\substack{S \subseteq N, |S|=i \\ j \in S}} \left(\bigcap_{\ell \in S} X_\ell = 1, \mu = i\right)\right) \\ &= \frac{1}{i} \sum_{j=1}^n \mathbb{P}(X_j = 1, \mu = i) \\ &= \sum_{j=1}^n \frac{1}{i} v_{ij}. \end{aligned}$$

Therefore,

$$\mathbb{P}(\mu \geq k) = \sum_{i=k}^n \sum_{j=1}^n \frac{1}{i} v_{ij}.$$

□

We scale v_{ij} by i so that the coefficient of v_{ij} , $\frac{1}{i}$, goes away, and then we rewrite

(101)-(104) as follows

$$\min : \sum_{i=k}^n \sum_{j=1}^n v_{ij} \quad (105)$$

$$\sum_{j=1}^n \sum_{i=0}^n v_{ij} = 1 \quad (106)$$

$$\sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j \quad t = 1, \dots, m \quad j = 1, \dots, n \quad (107)$$

$$v \geq 0 \quad i = 1, \dots, n \quad j = 1, \dots, n, \quad (108)$$

which is in a form more consistent with the expression of FAM in (82). After scaling, the equations in (100) become

$$i v_{ij} = \sum_{C: |C|=i, j \in C} v_C \quad i = 1, \dots, n \quad j = 1, \dots, n. \quad (109)$$

4.4.2 Strengthened PAM

The aggregation process in Section 4.4.1 is not complete until we project v_C s out of (109). We complete the projection and obtain extra inequalities that we use to tighten the model in (105)-(108). For simplicity, we perform projection for (100), and then make scaling on the resulting inequalities.

Proposition 23. *Let*

$$W_i = \{(\dots, v_C, \dots, v_{ij}, \dots) \in \mathbb{R}_+^{\binom{n}{i}} \times \mathbb{R}^n : v_{ij} = \sum_{C: |C|=i, j \in C} v_C \quad j = 1, \dots, n\}. \quad (110)$$

Then the projections of W_i s onto v_{ij} -space are the following sets:

$$Proj(W_i) = V_i = \{(v_{i1}, \dots, v_{in}) \in \mathbb{R}^n : -(i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0, v_{ij} \geq 0 \quad \forall j \in N\} \quad (111)$$

$$Proj(W_{n-1}) = V_{n-1} = \{(v_{(n-1)1}, \dots, v_{(n-1)n}) \in \mathbb{R}^n : -(n-2)v_{(n-1)j} + \sum_{t \neq j} v_{it} \geq 0, j \in N\} \quad (112)$$

and

$$Proj(W_n) = V_n = \{(v_{n1}, \dots, v_{nn}) \in \mathbb{R}_+^n : v_{n1} = v_{nt} \ 2 \leq t \leq n, \ v_{n1} \geq 0\}. \quad (113)$$

Proof. (1) We show that (111) holds for an arbitrary i between 2 and $n-1$, where $|C| = i$. To simplify notations, we drop subscript i in the constraint for now. Let $w = (\dots, v_C, \dots) \in \mathbb{R}^{\binom{n}{i}}$ and $v = (\dots, v_{ij}, \dots) \in \mathbb{R}^n$, we rewrite (110) as follows

$$W_i = \{(w, v) \in \mathbb{R}_+^{\binom{n}{i}} \times \mathbb{R}^n : v = Gw\},$$

where $G = (\dots, g_C, \dots) \in \mathbb{R}^{n \times \binom{n}{i}}$ is the coefficient matrix and $g_C = (g_C^1, \dots, g_C^j, \dots, g_C^n)^\top$ is the column corresponding to the variable v_C . $g_C^j = 1$ if $j \in C$; $g_C^j = 0$ if $j \notin C$. And G consists of all permutations of the 0-1 vector with i ones and $n-i$ zeros. Rewrite (111) as follows

$$V_i = \{v \in \mathbb{R}^n : (e - ie_j)^\top v \geq 0, \ e_j^\top v \geq 0 \ j = 1, \dots, n\},$$

where $e \in \mathbb{R}^n$ is a vector with all components equal to one and $e_i \in \mathbb{R}^n$ is the i -th unit vector.

To show that V_i is the projection of W_i into v -space, we need to show

$$\bar{v} \in V_i \Leftrightarrow \text{There is a } \bar{w} \in \mathbb{R}_+^{\binom{n}{i}} \text{ such that } G\bar{w} = \bar{v}.$$

By Farkas' lemma, the statement above is equivalent to the following: for any $u \in \mathbb{R}^n$ such that $u^\top G \geq 0$, $u^\top \bar{v} \geq 0$. Let $\{u_\ell\}$ be the set of extreme rays of the cone $\{u : u^\top G \geq 0\}$, it is sufficient to show that the set of constraint vectors in V_i , i.e., $(e - ie_j)$ and e_j , is exactly $\{u_\ell\}$.

We first show $(e - ie_j)$ and e_j are extreme rays.

$(e - ie_j)^\top$ is a feasible solution for the cone $\{u : u^\top G \geq 0\}$:

$$(e - ie_j)^\top g_C = \begin{cases} 0 & C : j \in C \\ i - 1 & C : j \notin C. \end{cases}$$

Furthermore, the products above have $\binom{n-1}{i-1}$ zeros, which means that $\binom{n-1}{i-1}$ constraints in $u^\top G \geq 0$ are tight. Notice that the binding constraint vectors g_C are all the permutations of vectors with j -th position fixed to one. Therefore, they span \mathbb{R}^{n-1} (see Lemma 5) and we can find $n - 1$ linearly independent vectors out of them which have zero products with $(e - ie_j)$. Thus, $(e - ie_j)$ is an extreme ray of $\{u : u^\top G \geq 0\}$.

As for e_j , we have $e_j^\top G$ as follows

$$e_j^\top g_C = \begin{cases} 1 & C : j \in C \\ 0 & C : j \notin C. \end{cases}$$

Therefore, e_j is a feasible solution for the cone $\{u : u^\top G \geq 0\}$ and the product above has $\binom{n-1}{i}$ zeros that corresponds to g_C with $j \notin C$. Among those columns that have zero products with e_j , we can always find $n - 1$ linearly independent ones since they span \mathbb{R}^{n-1} . Therefore, e_j is an extreme ray of $\{u : u^\top G \geq 0\}$.

Now we show that there are no other extreme rays. Suppose not, let λ be a new distinct extreme ray; let $g_{S_j} = (g_{S_j}^1, \dots, g_{S_j}^t, \dots, g_{S_j}^n)^\top$ $j = 1, \dots, n - 1$ be the set of linear independent columns of G with $\lambda^\top g_{S_j} = 0$ $j = 1, \dots, n - 1$. Since $\lambda \neq e_j$ for all $j = 1, \dots, n$, we have

$$\nexists t \text{ such that } g_{S_j}^t = 0 \quad j = 1, \dots, n - 1 \quad (114)$$

because otherwise e_t would be the extreme ray formed by the half planes $\{u^\top g_{S_j} \geq 0 \mid j = 1, \dots, n - 1\}$. Similarly, since $\lambda \neq (e - ie_j)$ for all $j = 1, \dots, n$, we have

$$\nexists t \text{ such that } g_{S_j}^t = 1 \quad j = 1, \dots, n - 1. \quad (115)$$

Because of (115) and $g_{S_j} \geq 0$, $\lambda_{t^*} < 0$ for some t^* (otherwise, $\lambda^\top g_{S_j} > 0$ for some j by (114)). By (114), there is a j^* such that $g_{S_{j^*}}^{t^*} = 0$. Since $\lambda^\top g_{S_{j^*}} = \sum_{t \neq t^*} \lambda_t g_{S_{j^*}}^t = 0$ and $g_{S_{j^*}} \neq 0$, we can find an index \bar{t} such that $\lambda_{\bar{t}} \geq 0$ and $g_{S_{j^*}}^{\bar{t}} = 1$. Let g^* be a vector obtained by switching the components at position t^* and \bar{t} in vector $g_{S_{j^*}}$. Note that g^* is also a vector of G . Then $\lambda^\top g^* = \lambda^\top g_{S_{j^*}} + \lambda_{t^*} - \lambda_{\bar{t}} < 0$. Thus, λ is not a feasible ray of the cone $\{u : u^\top G \geq 0\}$. Therefore, there are no extreme rays of $\{u : u^\top G \geq 0\}$ other than $\{(e - ie_j), e_i \mid j = 1, \dots, n\}$ and $\text{Proj}(W_i) = V_i$ for $i = 2, \dots, n-2$.

(2) For $\text{Proj}(W_{n-1})$, we can show $\{(e - ie_j) \mid j = 1, \dots, n\}$ are extreme rays of $\{u : u^\top G \geq 0\}$. But $\{e_j\}$ are not extreme ray in this case since each row of G has only one zero. With similar argument as in (2), we can show that $\{(e - ie_j) \mid j = 1, \dots, n\}$ are the only extreme rays and $\text{Proj}(W_n) = V_n$.

(3) Equations in (112) holds since $v_{nj} = v_N$ for all $j = 1, \dots, n$. □

Notice that the projection of W_1 yields only non-negativity constraints on v_{1j} $j = 1, \dots, n$ and the non-negativity of $v_{(n-1),j}$ for $j = 1, \dots, n$ are implied by $-(i-1)v_{(n-1),j} + \sum_{t \neq j} v_{it} \geq 0$ $j = 1, \dots, n$ (to see this, consider a multiplier $\lambda = (\frac{1}{n-1}, \dots, \frac{1}{n-1}, 0)$.) All the resulting inequalities from the projection remain unchanged after scaling.

We call the linear program (105)-(108) with additional constraints in (111), (112), and (113) as the *strengthened partially aggregated model* or *SPAM*.

Now we provide a proof for the result used in the proof of Proposition 23 in the following lemma.

Lemma 5. *Let $v \in \{0, 1\}^n$ has i ones and $n - i$ zeros, $0 < i < n$. Let $\{v_i\}$ be all the permutations of v . Then $\{v_i\}$ spans \mathbb{R}^n .*

Proof. We show that we can pick n vectors from $\{v_i\}$ to form a matrix M whose rank is n . Write M in a block-wise way:

$$\begin{bmatrix} M_1 & M_3 \\ M_2 & M_4 \end{bmatrix},$$

where $M_1 \in \mathbb{R}^{(i+1) \times (i+1)}$, $M_2 \in \mathbb{R}^{(n-i-1) \times (i+1)}$, and $M_4 \in \mathbb{R}^{(n-i-1) \times (n-i-1)}$. M_1 is the square matrix with all diagonal components equal to zero and the rest components equal to one. M_2 is a zero matrix. Clearly,

$$\begin{pmatrix} M_1 \\ M_2 \end{pmatrix} \subseteq \{v_i\}$$

and $\text{rank}(M_1) = i + 1$. M_4 is the $(n - i - 1) \times (n - i - 1)$ identity matrix and $\text{rank}(M_4) = n - i - 1$. M_3 is an $(i + 1) \times (n - i - 1)$ matrix with each column consisting of $(i - 1)$ ones and two zeros. Therefore,

$$\begin{pmatrix} M_3 \\ M_4 \end{pmatrix} \subseteq \{v_i\}$$

and the columns formed by M_3 and M_4 are independent from those formed by M_1 and M_2 . Thus, $\text{rank}(M) = \text{rank}(M_1) + \text{rank}(M_4) = n$. \square

The non-trivial inequalities in Proposition 23 are valid for formulation (105)-(108) and they are not implied by the constraints in PAM. We give some numerical examples in the next section to demonstrate the effectiveness of these inequalities in strengthening the lower bound produced by PAM.

4.4.3 Gaps Between PAM and FAM

We have introduced three linear programs that calculate lower and upper bounds for the probability of k -coverage: the Boolean model (BM), the fully-aggregated model (FAM), and the partially aggregated model (PAM) (i.e., the strengthened PAM is a version of PAM). Although the Boolean model provides the best possible bounds, it has an exponential number of variables, so it is not practical. The latter two models with a polynomial number of variables can be used to build tractable MILP formulations. Of the two, PAM provides no worse bounds than FAM, since FAM can be obtained by aggregating PAM. To numerically compare the bound strength

by PAM and FAM, we provide mathematical models that calculate maximal gap between them and conduct computational experiments. We restrict our attentions to the cases where only the first and second moments are available since this is the case in most practical situations.

4.4.3.1 *Model For Calculating the Maximal Gap*

Since the problem is to seek a probability distribution that maximizes the gap between PAM and FAM, marginal distributions p_i and pair-wise joint distributions p_{ij} become variables. s_1 and s_2 in FAM and s_1^j and s_2^j for every j in PAM become functions of p_i and p_{ij} , i.e., $s_1 = \sum_{i=1}^n p_i$, $s_2 = \sum_{i < j} p_{ij}$, $s_1^j = p_j$, and $s_2^j = \sum_{i \neq j} p_{ij}$ for all $j = 1, \dots, n$. Let z_{PAM}^* and z_{FAM}^* be the optimal lower bound on the probability of k -coverage calculated by PAM and FAM, respectively. Clearly, $z_{PAM}^* \geq z_{FAM}^*$. Let \mathcal{P}^2 be the set of all possible probability distributions with specified first and second

moments. The maximal gap can be calculated by the following optimization problem:

$$\begin{aligned}
\text{maximal gap} &= \max_{p \in \mathcal{P}^2} (z_{PAM}^* - z_{FAM}^*) \\
&= \max_{p \in \mathcal{P}^2} (\min_{v_{ij}} \{ \sum_{j=1}^n \sum_{i=k}^n v_{ij} : \sum_{j=1}^n \sum_{i=1}^n v_{ij} \leq 1, \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j \quad t = 1, 2, j = 1 \dots n, v_{ij} \geq 0 \} \\
&\quad - \min_{v_i} \{ \sum_{i=k}^n v_i : \sum_{i=1}^n v_i \leq 1, \sum_{i=t}^n \binom{i}{t} v_i = s_t \quad t = 1, 2, v_i \geq 0 \}) \\
&= \max_{p \in \mathcal{P}^2} (\max_{(\mu_0, \mu_t^j)} \{ \mu_0 + \sum_{j=1}^n \sum_{t=1}^2 \frac{1}{t} \mu_t^j s_t^j(p) : \mu_0 + \sum_{t=1}^2 \mu_t^j \binom{i}{t} \leq 0 \quad i = 1, \dots, k-1, \\
&\quad \mu_0 + \sum_{t=1}^2 \mu_t^j \binom{i}{t} \leq 1 \quad i = k, \dots, n, \mu_0 \leq 0 \} + \max_{v_i} \{ - \sum_{i=k}^n v_i : \sum_{i=1}^n v_i \leq 1, \sum_{i=t}^n \binom{i}{t} v_i = s_t \\
&\quad t = 1, 2, v_i \geq 0 \}) \\
&= \max_{p \in \mathcal{P}^2} (\max_{\mu_0, \mu_t^j, v_i} \{ \mu_0 + \sum_{j=1}^n \sum_{t=1}^2 \frac{1}{t} \mu_t^j s_t^j(p) - \sum_{i=k}^n v_i : \mu_0 + \sum_{t=1}^2 \mu_t^j \binom{i}{t} \leq 0 \quad i = 1, \dots, k-1, \\
&\quad \mu_0 + \sum_{t=1}^2 \mu_t^j \binom{i}{t} \leq 1 \quad i = k, \dots, n; \sum_{i=t}^n \binom{i}{t} v_i = s_t \quad t = 1, 2, \sum_{i=1}^n v_i \leq 1, v_i \geq 0, \mu_0 \leq 0 \}) \\
&= \max_{p \in \mathcal{P}^2, \mu_0, \mu_t^j, v_i} \mu_0 + \sum_{j=1}^n \sum_{t=1}^2 \frac{1}{t} \mu_t^j s_t^j(p) - \sum_{i=k}^n v_i \\
&\quad \mu_0 + \sum_{t=1}^2 \mu_t^j \binom{i}{t} \leq 0 \quad i = 1, \dots, k-1, \\
&\quad \mu_0 + \sum_{t=1}^2 \mu_t^j \binom{i}{t} \leq 1 \quad i = k, \dots, n; \\
&\quad \sum_{i=1}^n v_i \leq 1; \\
&\quad \sum_{i=t}^n \binom{i}{t} v_i = s_t \quad t = 1, 2, v_i \geq 0, \mu_0 \leq 0,
\end{aligned}$$

where μ_0 and μ_t^j are the dual variables associated with constraints (102) and (103). Let $p^1 := (\dots, p_i, \dots)$ and $p^2 := (\dots, p_{ij}, \dots)$, and $p = (p^1, p^2) \in [0, 1]^{n + \binom{n}{2}}$, set \mathcal{P}^2 can be mathematically defined as follows:

$$\mathcal{P}^2 = \{p \in [0, 1]^{n + \binom{n}{2}} : \exists (\Omega, \mathcal{F}, \mathbb{P}) \text{ s.t. } \mathbb{P}(A_i) = p_i \quad \forall i, \mathbb{P}(A_i \bigcap A_j) = p_{ij} \quad \forall i, j \quad i \neq j\},$$

where $A_i \quad i = 1, \dots, n$ are events in \mathcal{F} .

Since any probability distributions can be recovered by the Boolean problem using the probabilities of the mutually exclusive elementary events, the following proposition is straightforward.

Proposition 24.

$$\mathcal{P}^2 = P,$$

where

$$P = \{(p^1, p^2) \in \mathbb{R}_+^n \times \mathbb{R}_+^{\binom{n}{2}} : \exists v \in \mathbb{R}_+^{2^n} \text{ s.t. } \sum_{C \subseteq N} v_C \leq 1; \\ p_i = \sum_{i \in C} v_C \text{ } i \in N; p_{ij} = \sum_{i, j \in C, i < j} v_C\}. \quad (116)$$

\mathcal{P}^2 is the most general set. If we restrict our attention to the family of pair-wise independent probabilities, \mathcal{P}^2 can be expressed as follows:

$$P_I = \{(p^1, p^2) \in \mathbb{R}_+^n \times \mathbb{R}_+^{\binom{n}{2}} : p_{ij} = p_i * p_j \text{ } i, j \in N, i < j; p_i \leq 1 \text{ } i \in N\}. \quad (117)$$

In the next section, we optimize the gap over set P and P_I and give some numerical examples to compare the bound strength by PAM and FAM.

4.4.3.2 Numerical Examples

First, we present a set of examples with a small number of events represented by Bernoulli random variables. The distribution functions of the Bernoulli variables are obtained by the model in Section 4.4.3.1 optimizing the gap over set P defined in (116). Example 1 has four Bernoulli variables, namely, X_1, X_2, X_3 , and X_4 . The sample space has only two outcomes with probabilities 0.75 and 0.25. The outcomes and their probabilities are presented in Table 7

Table 7: Probability Distributions in Example 1

Outcome	Bernoulli Random Variables				Probability
	X_1	X_2	X_3	X_4	
1	1	1	0	0	0.75
2	1	1	1	1	0.25

With the mutually exclusive outcomes and their probabilities, we can immediately obtain marginal probabilities p_i and pair-wise joint probabilities p_{ij} and use

them as input data. Example 2 has five Bernoulli variables, and their outcomes and probabilities are presented in Table 8.

Table 8: Probability Distributions in Example 2

Outcome	Bernoulli Random Variables					Probability
	X_1	X_2	X_3	X_4	X_5	
1	1	0	0	0	0	5/6
2	1	1	1	1	1	1/6

Example 3 has six Bernoulli variables, and the outcomes and their probabilities are presented in Table 9. We calculate the lower bounds for the three examples using

Table 9: Probability Distributions in Example 3

Outcome	Bernoulli Random Variables						Probability
	X_1	X_2	X_3	X_4	X_5	X_6	
1	1	0	0	0	0	1	0.75
2	1	1	1	1	1	1	0.25

FAM, PAM, SPAM, and BM; and we compare the results in Table 10. When using SPAM, we only add constraints in (112), and the lower bounds produced are already as strong as those produced by BM.

Table 10: Comparison of Lower Bounds

Example	k	Lower Bounds			
		FAM	PAM	SPAM	BM
1	2	0.75	0.875	1	1
2	4	0	0.133	0.167	0.167
3	5	0	0.167	0.25	0.25

To empirically study the maximal gaps between bounds produced by FAM and PAM, we fix $n = 10$ and vary the values of k , and we use the model in Section 4.4.3.1 to maximize the gap between FAM and PAM over \mathcal{P}^2 defined in (116). We solve the optimization problems for ten hours, record the best possible solutions found (p_i and p_{ij}) and then use them as input data in the FAM and PAM model to calculate lower bounds on the probability of k -coverage. We calculate the gaps between lower bounds produced by FAM and PAM and present the results in Table 11. We also append the

Table 11: Comparison of Lower Bounds ($n=10$)

Model	k									
	1	2	3	4	5	6	7	8	9	10
FAM	0.360	0.438	0.063	0.042	0.063	0.100	0.357	0.000	0.000	0.000
PAM	1.000	1.000	0.625	0.756	0.844	0.900	0.918	0.200	0.300	0.467
GAP	0.640	0.563	0.563	0.714	0.781	0.800	0.561	0.200	0.300	0.467
SPAM	1.000	1.000	0.625	0.756	0.844	0.900	0.918	0.286	0.375	0.583
BM	1.000	1.000	0.625	0.756	0.844	0.900	0.918	0.286	0.375	0.583

lower bounds produced by SPAM and BM to Table 11 for comparison. Note that the “GAP” in Table 11 refers to the absolute gaps between the lower bounds produced by FAM and PAM, and they are not necessarily maximal since we enforce a time limit of ten hours on the solution. Table 11 shows that for each value of k , we can find a specific distribution function under which the gap between FAM and PAM can be significantly large. In this particular experiment, PAM produces lower bounds as good as those by SPAM and BM.

Then we restrict \mathcal{P}^2 to be family of independent distributions, i.e., $\mathcal{P}^2 = P_I$, where P_I is defined in (117), and calculate the maximal possible gap between FAM and PAM. A time limit of twenty hours is enforced on solving the optimization model in Section 4.4.3.1. The results are summarized in the Table 12. Comparing with the

Table 12: Comparison of Lower Bounds ($n=10$) (Independent Distributions)

Model	k									
	1	2	3	4	5	6	7	8	9	10
FAM	0.819	0.820	0.822	0.828	0.828	0.833	0.843	0.000	0.000	0.000
PAM	0.940	0.910	0.893	0.885	0.877	0.875	0.878	0.029	0.039	0.054
GAP	0.121	0.090	0.071	0.057	0.049	0.042	0.035	0.029	0.039	0.054
SPAM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.049	0.078	0.181
BM	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.049	0.078	0.181

gaps in Table 11, the gaps under the data independence assumption demonstrated in Table 12 tend to be significant smaller.

In the following, we use the instances in [100] and compare the lower bounds yield by SPAM with those presented in [100]. Notice that, the lower bounds in [100] are for the probability of union of events, i.e., $k = 1$. Examples 4, 5, and 6 have 20 Bernoulli

variables each and the outcomes and their probabilities are presented in Table 13, 14, and 15, respectively.

Table 13: Probability Distributions in Example 4

Outcome	Bernoulli Random Variables																				Probability
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	
1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0.012214
2	0	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0.022231
3	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0.023287
4	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0.033976
5	1	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1	0	0.034761
6	0	1	0	0	1	0	1	0	1	0	1	1	0	0	1	0	1	0	0	1	0.044582
7	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0.045943
8	0	1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0.055185
9	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0.056404
10	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0.066317
11	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0.067685
12	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0.077376
13	1	0	0	1	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	1	0.078648
14	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0.088878
15	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.292514

Table 14: Probability Distributions in Example 5

Outcome	Bernoulli Random Variables																				Probability
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	
1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	0	1	0	0.008964634
2	1	1	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0.02492217
3	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0.02109813
4	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	1	0	0	0.03779353
5	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0.0463261
6	1	0	0	0	0	0	1	0	1	1	1	1	1	0	1	0	0	0	0	1	0.04284324
7	1	1	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0.07804262
8	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	1	0.02536991
9	0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0.01916672
10	1	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0.06340085
11	0	1	0	1	1	0	0	0	1	1	0	1	0	0	1	1	0	0	0	1	0.07315289
12	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0.07732742
13	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0.0224802
14	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0.09164494
15	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0.3674566

We compare the lower bounds produced by SPAM with the results in [100] in Table 16. Note that all of the following bounds are calculated with only marginal probabilities and pair-wise joint probabilities, except those in the fourth column.

Table 15: Probability Distributions in Example 6

Outcome	Bernoulli Random Variables																				Probability
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}	
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0.1017688
2	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0.112992
3	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0.01514044
4	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0.05684733
5	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0.03270125
6	1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0.1005075
7	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0.07306695
8	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0.01743922
9	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0.06284498
10	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0.05830101
11	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0.06833096
12	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0.07153743
13	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0.04503293
14	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0.03487869
15	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0.1486024

Table 16: Comparison With Results in [100]

Example	FAM	PAM	PAM(3)	SPAM
4	0.8275266	0.8580833	0.8864460	0.9394167
5	0.8658182	0.9100646	0.9354100	0.9482229
6	0.8985498	0.9435812	0.9587778	0.9715460

The second column cites the results obtained by the formula derived in [44], which is equivalent to FAM. The third column cites the results obtained by the formula derived in [100], which is exactly PAM. The fourth column cites the results obtained by PAM but with three moments, including the triple-wise joint probabilities. Prékopa and Gao also developed heuristics to strengthen the lower bounds by PAM, but the best results they can obtain are no better than those obtained by involving triple-wise probabilities, which are listed in the fourth column. The fifth column gives the results obtained by SPAM. Table 16 demonstrates that given the same amount of information, SPAM makes the best use of the given information and produces by far the strongest lower bounds. On the contrary, since it misses out the extra constraints in (111)-(113), PAM does not make full use of the given information and could possibly produce a worse bound than SPAM even when provided with more information. This can be seen by comparing the fourth and fifth column. FAM makes the least use of

information and gives the worst bounds.

Now we use the same instances to calculate lower bounds for the probability of 3-coverage, i.e., $k = 3$, and summarize the results in Table 17, which shows again

Table 17: Lower Bounds for 3-Coverage				
Example	FAM	PAM	PAM(3)	SPAM
1	0.6643058	0.665249	0.6768353	0.6745504
2	0.7298830	0.7528989	0.7847514	0.8005835
3	0.7387907	0.7819380	0.8482093	0.8614323

that SPAM produces significantly tighter lower bounds than both FAM and PAM.

Notice that, when $k = 1$, the lower bounds produced by FAM and PAM have a neat closed form in the moments (s_t) and disaggregated moments (s_t^j) , respectively, which is useful for developing solution approaches; but SPAM does not have a straightforward closed form. When $k \geq 2$, none of the bounds produced by the three models have a closed-form expression.

4.4.4 Mixed-Integer Formulations

Following the same ideas in Section 4.3.1 and 4.3.2, we use PAM to build an inner approximation X_{SP} and an outer approximation X_{RP} and use SPAM to build an inner approximation X_{SS} and an outer approximation X_{RS} .

Proposition 25. *Define*

$$X_{RP} := \{x \in \{0, 1\}^n : \max\{\sum_{i=k}^n \sum_{j=1}^n v_{ij} : \sum_{i,j} v_{ij} = 1; \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j(x) \quad t = 1, \dots, m \quad j = 1, \dots, n, \\ v_{ij} \geq 0\} \geq 1 - \epsilon\}, \quad (118)$$

$$X_{SP} := \{x \in \{0, 1\}^n : \min\{\sum_{i=k}^n \sum_{j=1}^n v_{ij} : \sum_{i,j} v_{ij} = 1; \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j(x) \quad t = 1, \dots, m \quad j = 1, \dots, n, \\ v_{ij} \geq 0\} \geq 1 - \epsilon\}, \quad (119)$$

$$X_{RS} := \{x \in \{0, 1\}^n : \max\{\sum_{i=k}^n \sum_{j=1}^n v_{ij} : \sum_{i,j} v_{ij} = 1; \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j(x) \quad t = 1, \dots, m \quad j = 1, \dots, n, \\ - (i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0 \quad j = 1, \dots, n, \quad i = 2, \dots, n-1, \quad v_{n1} = v_{nt} \quad 2 \leq t \leq n, \quad v_{ij} \geq 0\} \geq 1 - \epsilon\}, \quad (120)$$

and

$$X_{SS} := \{x \in \{0, 1\}^n : \min\{\sum_{i=k}^n \sum_{j=1}^n v_{ij} : \sum_{i,j} v_{ij} = 1; \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} s_t^j(x) \quad t = 1, \dots, m \quad j = 1, \dots, n, \\ - (i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0 \quad j = 1, \dots, n, \quad i = 2, \dots, n-1, \quad v_{n1} = v_{nt} \quad 2 \leq t \leq n, \quad v_{ij} \geq 0\} \geq 1 - \epsilon\}.$$
(121)

Then, X_{RP} and X_{RS} are relaxations of X_B ; X_{SP} and X_{SS} are restrictions of X_B .

The above sets can be linearized with the McCormick linearization technique as follows

Proposition 26.

$$X_{RP} = \{x \in \{0, 1\}^n : \sum_{i=1}^n \sum_{j=1}^n v_{ij} \leq 1 \\ \sum_{i=k}^n \sum_{j=1}^n v_{ij} \geq 1 - \epsilon \\ \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} \sum p_{C_j}^t y_{C_j}^t \quad t = 2, \dots, m \quad j = 1, \dots, n \quad \} \quad (122)$$

$$y_{C_j}^t \leq x_i \quad \forall i \in C_j \quad t = 1, \dots, m \quad j = 1, \dots, n$$

$$y_{C_j}^t \geq 1 + (\sum_{i \in C_j} x_i - t) \quad \forall C_j \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n$$

$$v_{ij} \geq 0 \quad \forall i, j; \quad y_{C_j}^t \geq 0 \quad \forall C_j \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n$$

and

$$v_0 + \sum_{i=1}^n \sum_{j=1}^n v_{ij} = 1 \\ \sum_{i=k}^n \sum_{j=1}^n v_{ij} \geq 1 - \epsilon \\ \sum_{i=t}^n \binom{i}{t} v_{ij} = \frac{1}{t} \sum p_{C_j}^t y_{C_j}^t \quad t = 2, \dots, m \quad j = 1, \dots, n \\ X_{RS} = \{x \in \{0, 1\}^n : - (i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0 \quad j = 1, \dots, n, \quad i = 2, \dots, n-1, \quad \} \quad (123)$$

$$v_{n1} = v_{nt} \quad 2 \leq t \leq n,$$

$$y_{C_j}^t \leq x_i \quad \forall i \in C_j \quad t = 1, \dots, m \quad j = 1, \dots, n$$

$$y_{C_j}^t \geq 1 + (\sum_{i \in C_j} x_i - t) \quad \forall C_j \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n$$

$$v_{ij} \geq 0 \quad \forall i, j; \quad y_{C_j}^t \geq 0 \quad \forall C_j \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n$$

where $\mathcal{I}_j^t := \{I \subseteq N : |I| = t, j \in I\}$.

Let π_0 and π_i^j be the dual variables corresponding to the constraints (106) and (107). Let l_t^j and u_t^j be lower and upper bounds for π_t^j , respectively, we define $M_t^{j-} = \min\{0, l_t^j\}$ and $M_t^{j+} = \max\{0, u_t^j\}$. Then

$$\begin{aligned}
& \pi_0 + \sum_{i=1}^m \binom{t}{i} \pi_i^j \leq e_t \quad t = 1, \dots, n; \quad j = 1, \dots, n \\
& \pi_0 + \sum_{j=1}^n \sum_{t=1}^m \frac{1}{t} p_{C_j^t} y_{C_j^t} \geq 1 - \epsilon \\
& y_{C_j^t} \leq M_t^{j+} x_i \quad \forall i \in C_j^t \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
& X_{SP} = \{x \in \{0, 1\}^n : y_{C_j^t} \geq M_t^{j-} x_i \quad \forall i \in C_j^t \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \quad \}, \quad (124) \\
& y_{C_j^t} \leq \pi_t^j - M_t^{j-} (t - \sum_{i \in C_j^t} x_i) \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
& y_{C_j^t} \geq \pi_t^j - M_t^{j+} (t - \sum_{i \in C_j^t} x_i) \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
& \pi_0 \leq 0; \quad \pi_i^j \text{ free } \forall i, j; \quad y_{C_j^t} \geq 0 \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n
\end{aligned}$$

where e_t is a scalar: $e_t = 0$ if $t < k$; $e_t = 1$, otherwise.

Let μ_i^j be the dual variable corresponding to the constraint $-(i-1)v_{ij} + \sum_{t \neq j} v_{it} \geq 0$ in (121) for $i = 2, \dots, n-1$ and $j = 1, \dots, n$, and μ_n^j be the dual variable corresponding to the constraint $v_{n1} = v_{nj}$ in (121) for $j = 2, \dots, n$.

$$\begin{aligned}
\pi_0 + \pi_1^j &\leq e_1 \quad j = 1, \dots, n \\
\pi_0 + \sum_{i=1}^m \binom{t}{i} \pi_i^j - (t-1)\mu_j^t + \sum_{\ell \neq j} \mu_t^\ell &\leq e_t \quad t = 2, \dots, n-1; \quad j = 1, \dots, n \\
\pi_0 + \sum_{i=1}^m \binom{n}{i} \pi_i^1 - \sum_{j=2}^n \mu_n^j &\leq e_n \\
\pi_0 + \sum_{i=1}^m \binom{n}{i} \pi_i^j + \mu_n^j &\leq e_n \quad j = 2, \dots, n \\
\pi_0 + \sum_{j=1}^n \sum_{t=1}^m \frac{1}{t} p_{C_j^t} y_{C_j^t} &\geq 1 - \epsilon
\end{aligned}$$

$$X_{SS} = \{x \in \{0, 1\}^n : y_{C_j^t} \leq M_t^{j+} x_i \quad \forall i \in C_j^t \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \quad \}. \quad (125)$$

$$\begin{aligned}
y_{C_j^t} &\geq M_t^{j-} x_i \quad \forall i \in C_j^t \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
y_{C_j^t} &\leq \pi_t^j - M_t^{j-} (t - \sum_{i \in C_j^t} x_i) \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
y_{C_j^t} &\geq \pi_t^j - M_t^{j+} (t - \sum_{i \in C_j^t} x_i) \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
\pi_0 &\leq 0; \quad \pi_i^j \text{ free } \forall i, j; \quad y_{C_j^t} \geq 0 \quad \forall C_j^t \in \mathcal{I}_j^t \quad t = 1, \dots, m \quad j = 1, \dots, n \\
\mu_i^j &\geq 0 \quad j = 1, \dots, n, i = 2, \dots, n-1; \quad \mu_n^j \text{ free } j = 2, \dots, n
\end{aligned}$$

4.4.5 Bounds for Dual Extreme Points

In this section, we analyze the dual extreme solutions in PAM and derive bounds for dual variables π_t^j , to determine appropriate values for M_t^{j-} and M_t^{j+} used in formulation (124). In the following result, we discuss the case when $m = 2$ because marginal probabilities and pair-wise joint probabilities are often the only available information in practice.

Proposition 27. *Let the dual variables associated with the constraints in (102)-(104) be π_0 and π_i^j $i = 1, \dots, n$ and $j = 1, \dots, n$. Then*

1. *when $k = 1$:*

$$\pi_0 = 0, \pi_1^j \in [\frac{2}{n}, 1], \text{ and } \pi_2^j \in [-1, -\frac{2}{n(n-1)}] \quad j = 1, \dots, n. \quad (126)$$

2. when $k = 2$:

$$\pi_0 \in [-2, 0], \pi_1^j \in [0, 2], \text{ and } \pi_2^j \in [-1, \frac{2}{n(n-1)}] \quad j = 1, \dots, n. \quad (127)$$

3. when $k \geq 3$:

$$\begin{aligned} \pi_0 \in [-\frac{(k-1)(k+2)}{2}, 0], \pi_1^j \in [-\frac{k-2}{n(n-k+1)}, \frac{(k-1)(k+2)}{2}], \\ \text{and } \pi_2^j \in [-\frac{(k-1)(k+2)}{2}, \frac{2}{n(n-k+1)}] \quad j = 1, \dots, n. \end{aligned} \quad (128)$$

Proof. Following the same arguments in Proposition 21, we examine all possible extreme points of the dual polyhedron in the below:

$$\pi_0 \leq 0 \quad (129)$$

$$\pi_0 + i\pi_1^j + \binom{i}{2}\pi_2^j \leq 0 \quad i = 1, \dots, k-1, \quad j = 1, \dots, n \quad (130)$$

$$\pi_0 + i\pi_1^j + \binom{i}{2}\pi_2^j \leq 1 \quad i = k, \dots, n, \quad j = 1, \dots, n. \quad (131)$$

First, it is not clear that the dual polyhedron is full-dimensional in $(2n+1)$ -space. Second, the constraints (130) and (131) can be grouped by superscript j into n blocks, each of which consists of n constraints, and the constraint (129) can be grouped into any block (w.l.o.g. we assume (129) is consisted in Block 1, i.e., the set of constraints with $j = 1$). Each block has only two variables π_1^j and π_2^j and a common variable π_0 that is shared by all blocks. Notice that any three constraints in a block are linearly independent. Therefore, at an extreme point, Block 1 has at least three constraints that are tight and we call the indices of three tight constraints a *basis*. Each of the rest of the blocks has at least two constraints tight. Without loss of generality, we assume Block 2 with variables $(\pi_0, \pi_1^2, \pi_2^2)$ has at least two constraints that are tight. Since the n blocks of constraints are identical, it is sufficient to consider the possible values taken by the variables in Blocks 1 and 2. A basic feasible solution is a combination of the values $(\pi_0, \pi_1^1, \pi_2^1)$ and (π_1^2, π_2^2) .

We first apply the result in Proposition 21 to Block 1 and obtain:

$$\pi_0 \in [-\frac{(k-1)(k+2)}{2}, 0], \pi_1^1 \in [-\frac{k-2}{n(n-k+1)}, k] \text{ and } \pi_2^1 \in [-1, \frac{2}{n(n-k+1)}]. \quad (132)$$

There are two possible cases for the values of π_0 : $\pi_0 = 0$ or $\pi_0 = -\frac{(k-1)(2i-k+2)}{(i-k+1)(i-k+2)}$ $i = k, \dots, n-1$. When π_0 is fixed at a certain value, there are two constraints that are tight in Block 2 at an extreme point. With slight ambiguity, we call the index set $I \subseteq \{1, \dots, n\}$ of these tight constraints in block 2 as *sub-basis*. We discuss the possible values for π_1^2 and π_2^2 according to the values of π_0 .

1. $\pi_0 = 0$: There are two constraints in Block 2 that are tight and $\pi_0 = 0$. Hence, Block 2 yields the same solutions as those by Block 1 with $\pi_0 = 0$. Therefore, in this case π_1^2 and π_2^2 coincide with values of π_1^1 and π_2^1 with $\pi_0 = 0$.
2. $\pi_0 = -\frac{(k-1)(2i-k+2)}{(i-k+1)(i-k+2)}$, $i = k, \dots, n-1$:

The possible sub-bases are $I_s = \{t, t+1\}$, where $t \in \{1, k-2\} \cup \{i+1, \dots, n-1\}$ if $k \geq 3$, or $t \in \{i+1, \dots, n-1\}$ if $k = 2$ (see Lemma 6). Notice that, when $k = 1$, $\pi_0 = 0$ by Proposition 21.

When $k \geq 3$:

- (a) When $I_s \subseteq \{1, k-1\}$, we solve the following equation systems with π_0 fixed at $-\frac{(k-1)(2i-k+2)}{(i-k+1)(i-k+2)}$

$$\begin{aligned} \pi_0 + t\pi_1^2 + \binom{t}{2}\pi_2^2 &= 0 \\ \pi_0 + (t+1)\pi_1^2 + \binom{t+1}{2}\pi_2^2 &= 0. \end{aligned}$$

- (b) When $I_s \subseteq \{i+1, n\}$, we solve the following equation systems with π_0 fixed at $-\frac{(k-1)(2i-k+2)}{(i-k+1)(i-k+2)}$

$$\begin{aligned} \pi_0 + t\pi_1^2 + \binom{t}{2}\pi_2^2 &= 1 \\ \pi_0 + (t+1)\pi_1^2 + \binom{t+1}{2}\pi_2^2 &= 1. \end{aligned}$$

We summarize the solutions to the above two systems and write π_1^2 and π_2^2 as functions of i and t :

$$\pi_1^2(i, t) = \begin{cases} \frac{2(2i-k+2)(k-1)}{(i-k+1)(i-k+2)(t+1)} & t \in \{1, \dots, k-2\}, i \in \{k, \dots, n-1\}; \\ \frac{2i(i+1)}{(i-k+1)(i-k+2)(t+1)} & t \in \{i+1, \dots, n-1\} i \in \{k, \dots, n-2\}. \end{cases}$$

$$\pi_2^2(i, t) = \begin{cases} -\frac{2(2i-k+2)(k-1)}{(i-k+1)(i-k+2)(t+1)t} & t \in \{1, \dots, k-2\}, i \in \{k, \dots, n-1\}; \\ -\frac{2i(i+1)}{(i-k+1)(i-k+2)(t+1)t} & t \in \{i+1, \dots, n-1\} i \in \{k, \dots, n-2\}. \end{cases}$$

It is not hard to see that, for a fixed i , $\pi_1^2(i, t)$ is monotonously decreasing on $t \in \{1, k-2\}$ and $t \in \{i+1, \dots, n-1\}$; Furthermore, $\pi_1^2(i, 1) \geq \pi_1^2(i+1, 1)$ and $\pi_1^2(i, i+1) \geq \pi_1^2(i+1, i+2)$. Therefore,

$$\max_{i,t} \{\pi_1^2(i, t)\} = \max\{\pi_1^2(k, 1), \pi_1^2(k, k+1)\} = \pi_1^2(k, 1) = \frac{(k-1)(k+2)}{2}. \quad (133)$$

As for the minimal value of $\pi_1^2(i, t)$, we know that $\pi_1^2(i, k-2) \geq \pi_1^2(i+1, k-2)$ and $\pi_1^2(i, n-1) \geq \pi_1^2(i+1, n-1)$. Therefore,

$$\begin{aligned} \min_{i,t} \{\pi_1^2(i, t)\} &= \min\{\pi_1^2(n-1, k-2), \pi_1^2(n-2, n-1)\} \\ &= \min\left\{\frac{2(2n-k)}{(n-k)(n-k+1)}, \frac{2(n-1)(n-2)}{n(n-k-1)(n-k)}\right\} \geq 0. \end{aligned} \quad (134)$$

Similarly, $\pi_2^2(i, t)$ is monotonously increasing on $t \in \{1, k-2\}$ and $t \in \{k, \dots, n-2\}$. Furthermore, $\pi_2^2(i, 1) \leq \pi_2^2(i+1, 1)$ and $\pi_2^2(i, i+1) \leq \pi_2^2(i+1, i+2)$. Therefore,

$$\min_{i,t} \{\pi_2^2(i, t)\} = \min\{\pi_2^2(k, 1), \pi_2^2(k, k+1)\} = \pi_2^2(k, 1) = -\frac{(k-1)(k+2)}{2}. \quad (135)$$

As for the maximal value of $\pi_2^2(i, t)$, we know that $\pi_2^2(i, k-2) \leq \pi_2^2(i+1, k-2)$ and $\pi_2^2(i, n-1) \leq \pi_2^2(i+1, n-1)$. Therefore,

$$\begin{aligned} \max_{i,t} \{\pi_2^2(i, t)\} &= \max\{\pi_2^2(n-1, k-2), \pi_2^2(n-2, n-1)\} \\ &= \max\left\{-\frac{2(2n-k)}{(k-2)(n-k)(n-k+1)}, -\frac{2(n-2)}{n(n-k-1)(n-k)}\right\} \leq 0. \end{aligned} \quad (136)$$

When $k = 2$:

In this case, $t \in \{i + 1, \dots, n - 1\}$. Therefore,

$$\max_{i,t} \{\pi_1^2(i, t)\} = \pi_1^2(k, k + 1) = \frac{k(k + 1)}{(k + 2)} = \frac{3}{2}. \quad (137)$$

$$\min_{i,t} \{\pi_1^2(i, t)\} = \pi_1^2(n - 2, n - 1) = \frac{2(n - 1)}{n(n - 3)}. \quad (138)$$

and

$$\min_{i,t} \pi_2^2(i, t) = \pi_2^2(k, k + 1) = -\frac{k}{k + 2} = -\frac{1}{2}, \quad (139)$$

$$\max_{i,t} \pi_2^2(i, t) = \pi_2^2(n - 2, n - 1) = -\frac{2}{n(n - 3)}. \quad (140)$$

Since the constraint $\pi_0 \leq 0$ can also be grouped into Block 2, π_1^2 and π_2^2 will take the same values as π_1^1 and π_2^1 . Therefore, we summarize (137) - (140) and (91) by (127); we summarize (133)-(136) by (128); and for the case when $k = 1$, (126) is the same as (90).

□

The following lemma discusses feasible sub-bases in Block 2 with respect to the bases in Block 1.

Lemma 6. *Let I and I_s be the set of indices of the feasible basis in block 1 and the set of indices of sub-basis in block 2, respectively. Then*

1. *when $I \in \{0, 1, \dots, k - 1\}$ or $I = \{0, k - 1, n\}$, $I_s \in \{1, \dots, k - 1\}$ or $I = \{k - 1, n\}$.*
2. *when $I = \{k - 1, i, i + 1\}$, $I_s = \{t, t + 1\}$ where $t \in \{1, \dots, k - 2\}$ or $t \in \{k + 1, \dots, n - 1\}$.*

Proof. Once I is determined, the value of π_0 can be obtained by solving the equations with indices I . Note that π_0 is also a variable in the constraints of Block 2, i.e.,

$$i\pi_1^2 + \binom{i}{2}\pi_2^2 \leq e_i - \pi_0 \quad i = 1, \dots, n, \quad (141)$$

where $e_i = 0$ if $i \in \{1, \dots, k-1\}$; $e_i = 1$ if $i \in \{k, \dots, n\}$. Denote (141) as $G\pi \leq h$, where $\pi = (\pi_1^2, \pi_2^2)^\top$. For a basic feasible solution π , we have

$$G_{I_s}\pi = h_{I_s}$$

$$G_{I_r}\pi \leq h_{I_r}.$$

where G_{I_s} consists of the row vectors of (141) with indices in I_s , and $I_r = N \setminus I_s$. Solving $G_{I_s}\pi = h_{I_s}$, we have $\pi = G_{I_s}^{-1}h_{I_s}$, which is feasible for the constraints in I_r , i.e.,

$$G_{I_r}G_{I_s}^{-1}h_{I_s} \leq h_{I_r} \Rightarrow h_{I_r}^\top (G_{I_s}^\top)^{-1} G_{I_r}^\top \leq h_{I_r}^\top.$$

The above system is equivalent to the following statement, for all $p \in I_r$,

$$h_{I_r}^\top (G_{I_s}^\top)^{-1} a_p^\top \leq h_p \Rightarrow h_p - h_{I_r}^\top (G_{I_s}^\top)^{-1} a_p^\top \geq 0,$$

which can be written as the following vector

$$\begin{bmatrix} 1 & h_{I_s}^\top \\ 0 & G_{I_s}^\top \end{bmatrix}^{-1} \begin{bmatrix} h_p \\ a_p^\top \end{bmatrix}$$

with the first component being non-negative. And this vector is a solution to the following system of equations:

$$\begin{bmatrix} 1 & h_{I_s}^\top \\ 0 & G_{I_s}^\top \end{bmatrix} \begin{bmatrix} z_p \\ g_p \end{bmatrix} = \begin{bmatrix} h_p \\ a_p^\top \end{bmatrix}.$$

By the Cramer's rule, $z_p = \frac{|A|}{|B|}$, where $B = \begin{bmatrix} 1 & h_{I_s}^\top \\ 0 & G_{I_s}^\top \end{bmatrix}$ and $A = \begin{bmatrix} h_p & h_{I_s}^\top \\ a_p^\top & G_{I_s}^\top \end{bmatrix}$.

Therefore, for I_s to be a feasible sub-basis, z_p must be nonnegative for every $p \in N \setminus I_s$.

1. When $I = \{k-1, i, i+1\}$: We claim that $I_s = \{t, t+1\}$, where $t \in \{1, \dots, k-2\}$ when $k \geq 3$ or $t \in \{k+1, \dots, n-1\}$. Suppose indices of the sub-bases are not

consecutive pairs. Let $I = \{j, t\}$ and $t - j \geq 2$. Then, there is a p such that $j < p < t$. And

$$B = \begin{bmatrix} 1 & e_j & e_t \\ 0 & j & t \\ 0 & \binom{j}{2} & \binom{t}{2} \end{bmatrix} \text{ and } A = \begin{bmatrix} e_p - \pi_0 & e_j - \pi_0 & e_t - \pi_0 \\ p & j & t \\ \binom{p}{2} & \binom{j}{2} & \binom{t}{2} \end{bmatrix}.$$

We show that there exists a $p \in N \setminus I_s$ such that $z_p < 0$.

Since $|B| = \frac{ij(t-j)}{2} > 0$, $z_p < 0$ if $|A| < 0$.

$$|A| = \begin{vmatrix} e_p & e_j & e_t \\ p & j & t \\ \binom{p}{2} & \binom{j}{2} & \binom{t}{2} \end{vmatrix} - \pi_0 \begin{vmatrix} 1 & 1 & 1 \\ p & j & t \\ \binom{p}{2} & \binom{j}{2} & \binom{t}{2} \end{vmatrix}.$$

Denote the right-hand side item above as $\delta_1 - \pi_0 \delta_2$. Notice that, $\pi_0 < 0$ and $\delta_2 = \frac{(j-p)(j-t)(p-t)}{2} < 0$. The values of (e_p, e_j, e_t) depend on the values of p, j , and t :

- (a) When $j < t < k$: $(e_p, e_j, e_t) = (0, 0, 0)$. $\delta_1 < 0$ and $z_p < 0$;
- (b) When $k \leq j < t \leq n$: $(e_p, e_j, e_t) = (1, 1, 1)$. $\delta_1 < 0$ and $z_p < 0$;
- (c) When $j < k - 1$ and $t \geq k$: consider $j < p \leq k - 1$, then $(e_p, e_j, e_t) = (0, 0, 1)$. $\delta_1 = jp(j-p)/2 < 0$ and $z_p < 0$;
- (d) When $j \leq k - 1$: consider a $j \geq k$, then $(e_p, e_j, e_t) = (1, 0, 1)$. $\delta_1 = j(j-p-t)(p-t)/2 < 0$ and $z_p < 0$.

Therefore, we conclude that I_s must be consecutive pairs, i.e., $I_s = \{t, t+1\}$.

Now let us determine the range of t .

- (a) When $t \geq i + 1$: consider an arbitrary $p \in I_r$: if $e_p = 1$, then $\delta_1 = \delta_2 = (t-p)(t+1-p)/2 > 0$; if $e_p = 0$, $\delta_1 - \pi_0 \delta_2 > 0$ and $z_p > 0$;
- (b) When $t \leq k - 2$: consider an arbitrary $p \in I_r$: if $e_p = 1$, then $|A| = (t+t^2)/2 > 0$; if $e_p = 0$, then $|A| > 0$. Therefore, $z_p > 0$;

(c) When $t \in \{k-1, \dots, i\}$: consider $p = k-1$, then $e_p = 0$.

$|A| = \frac{(-1+k)(i-t)((-2+k)(1+t)+i(k-2(1+t)))}{2(1+i-k)(2+i-k)}$. Notice that the denominator is positive. The numerator is a linear function of t on $\{k-1, \dots, i\}$, i.e., $f(t) = (-1+k)(i-t)((-2+k)(1+t)+i(k-2(1+t)))$. Since $f(k-1) < 0$ and $f(i) < 0$, $f(t) < 0$ on $\{k-1, \dots, i\}$. Therefore, $|A| < 0$ and $z_p < 0$.

We conclude that $t \in \{1, k-2\} \cup \{i+1, \dots, n-1\}$ in this case.

2. When $I \in \{0, \dots, k-1\}$ or $I = \{0, k-1, n\}$. Note that $\pi_0 = 0$ in this case.

When $\pi_0 = 0$, considering the fact that the coefficient matrices in Blocks 1 and 2 are identical, the feasible sub-bases in block 2 coincide with the feasible basis in Block 1 when $\pi_0 = 0$, which is $I_s \in \{1, \dots, k-1\}$ or $I_s = \{k-1, n\}$

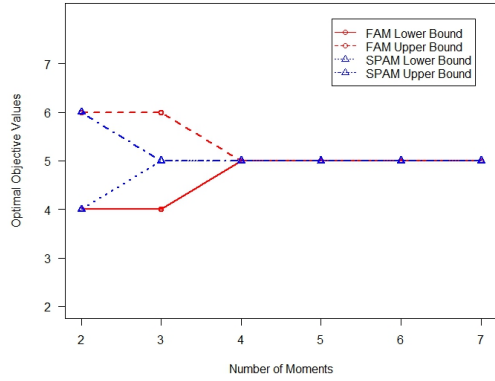
□

Since the dual polyhedron of SPAM does not have an organized structure, we do not attempt to analyze appropriate values for M s used in formulation (125). Nevertheless, formulation (125) will be a valid restriction even if M is not sufficiently large. Therefore, formulation (125) can be solved in a heuristic manner: we start with a moderately large value for M and solve the MIP; if $y_{C_j^t} \neq \pi_t^j$ for some i and j , and $y_{C_j^t}$ is at one of its bounds M_t^{j+} or M_t^{j-} , then double the tight bound and resolve the MIP till we are satisfied by the solution. For our instances, moderately large values for M produce solutions that are good enough in the first round. Therefore, we do not solve (125) iteratively.

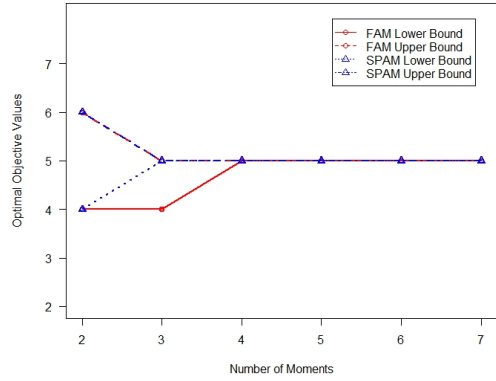
4.4.6 Numerical Examples

To study the improvement on the approximation quality by stronger probability bound results, we conduct computational experiments on the same instances used in Section 4.3.3 and compare upper and lower bounds on the objective costs.

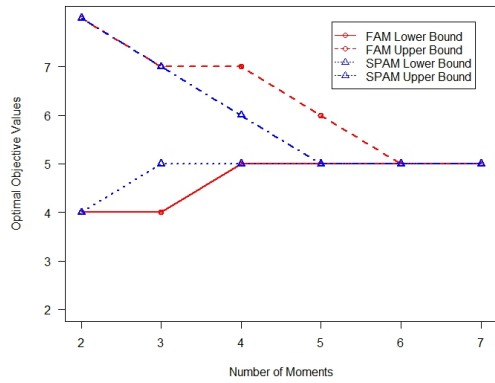
We use PAM-based MIP formulations, (122) and (124), and SPAM-based MIP formulations, (123) and (125), to calculate lower and upper bounds for the problems tested in Section 4.3.3. Then we compare with the bounds by BM and FAM-based MIP formulations. We select the four instances studied in Figure 2 and illustrate the comparisons in the figures in Figure 4. Note that, we do not plot the bounds by BM and PAM in the figures, because for each instance and each value of m , the PAM-based formulations produce the same bounds as the FAM-based formulations; the SPAM-based formulations produce the same bounds as the BM-based formulations. Therefore, the curves for FAM would coincide with those for PAM, and the curves for BM would coincide with those for SPAM.



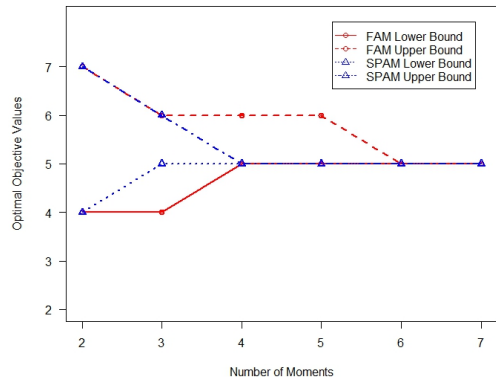
(a) Instance A



(b) Instance B



(c) Instance C



(d) Instance D

Figure 4: Comparison of Bounds by FAM and SPAM

The comparison shows that tighter probability bounds can help improve the quality of inner and outer approximations. In this experiment, the SPAM-based approximations can be accurate for each value of m . However, the PAM-based approximations produce no better bounds than the FAM-based approximations. This result can be explained by the fact that, for the randomly generated distributions in this experiment, bounds on the 3-coverage probability obtained by SPAM are as strong as those by BM; bounds obtained by PAM are as strong as those by FAM. Among the three tractable formulations based on FAM, PAM, and SPAM, the SPAM-based formulations provide the most precise over and under estimations for the true objective

value; however, they have the largest formulation size among the three. We remark that the above observations are not conclusive since they are based on the particular family of distributions we generated. For a specific application, the tradeoff between model precision and computational difficulty should be considered. For example, the quality of bounds on probability by different models (FAM, PAM, and SPAM), given the specific distributions, should be tested and compared before choosing one to develop MIP approximations.

4.5 *Applications of Modeling Techniques*

The probabilistic k -cover model is in essence to calculate the probability that a number of events occur, which appears in many real world applications. Besides the sensor deployment example, which is a straightforward application, we introduce a few more applications in this section to demonstrate the modeling capability of the probabilistic k -cover model.

4.5.1 Discrete Distributions With Non-uniform Upper Bounds

Bernoulli random number \tilde{a}_j in our problems can be easily extended to random number $\tilde{a}'_j = t_j \times \tilde{a}_j$ for all j , where t_j is a positive integer. Our model is valid with appropriate scaling if t_j is the same value for all j . In a more general case, where t_j s are not equal, the left-hand side is no longer a random variable that represents the number of events occurred, but the summation of n random number with different possible values. Therefore, the probabilistic k -cover model cannot be applied in a straightforward manner. However, since the data still exhibit a Bernoulli-like nature, we could treat the discrete distribution with nonuniform upper bounds as a set of Bernoulli random variables by duplicating the Bernoulli random variables t_j times.

Consider the following covering problem

$$\sum_{j=1}^n A_j x_j \geq k, \quad (142)$$

where A_j is a random number taking only two values: 0 and t_j ($t_j > 0$) with $\mathbb{P}(A_j = t_j) = p_j$ and $\mathbb{P}(A_j = 0) = 1 - p_j$. We use the product of t_j with a Bernoulli random number to represent A_j , i.e., $A_j = t_j \tilde{a}_j$. $\mathbb{P}(\tilde{a}_j = 1) = \mathbb{P}(A_j = t_j) = p_j$ and $\mathbb{P}(\tilde{a}_{j_1} = 1 \wedge \tilde{a}_{j_2} = 1) = \mathbb{P}(A_{j_1} = t_{j_1} \wedge A_{j_2} = t_{j_2}) = p_{j_1 j_2}$.

We define new Bernoulli random variables by duplicating \tilde{a}_j : $\tilde{a}_j^i = \tilde{a}_j$, $i=1, \dots, t_j$ for each $j \in N$. Since the \tilde{a}_j^i s are duplications of the original Bernoulli variable \tilde{a}_j , $\sum_{i=1}^{t_j} \tilde{a}_j^i$ is a random variable with only two values: 0 and t_j , and $\mathbb{P}(\sum_{i=1}^{t_j} \tilde{a}_j^i = t_j) = p_j$ and $\mathbb{P}(\sum_{i=1}^{t_j} \tilde{a}_j^i = 0) = 1 - p_j$. Clearly, the joint probability for duplicated random variables originating from the same j , $\mathbb{P}(\tilde{a}_j^{i_1} \wedge \tilde{a}_j^{i_2})$, equals $\mathbb{P}(\tilde{a}_j)$, and the joint probability for duplicated random variables originating from different j , $\mathbb{P}(\tilde{a}_{j_1}^i \wedge \tilde{a}_{j_2}^i)$, equals $\mathbb{P}(\tilde{a}_{j_1} \wedge \tilde{a}_{j_2})$. Higher-order joint probabilities can be set up in a similar way. Therefore, the random variable A_j in (142) can be replaced by $\sum_{i=1}^{t_j} \hat{a}_j^i$:

$$\sum_{j=1}^n \left(\sum_{i=1}^{t_j} \hat{a}_j^i \right) x_j \geq k,$$

or equivalently

$$\sum_{j=1}^n \sum_{i=1}^{t_j} \hat{a}_j^i x_j \geq k. \quad (143)$$

The left-hand side of inequality (143) is the summation of a set of Bernoulli random variables parameterized by x , and the inequality reads that at least k out of the $\sum_{j \in N} t_j$ Bernoulli events, $\{\hat{a}_j^i x_j = 1\}$, occur. Therefore, this problem can be treated as a probabilistic set k -cover problem.

4.5.2 Location Problems

The maximum availability location problem is studied in a probabilistic setting in [101]. The problem is to place p facilities, such as service vehicles and fire stations, on a set J of possible locations in a region divided into a set I of sub-regions. A sub-region $i \in I$, with a certain population f_i , is considered covered if a call is answered by a fire station, or if a service vehicle is within a certain range with the prescribed

reliability of α . The goal is to find the allocation of the facilities that maximizes the total population covered by at least one facility. The probability that a facility is not available is estimated by the busy fraction r . Based on the assumption that the busy fractions are independent across all facilities, the probabilistic constraint $\mathbb{P}\{\text{one or more facilities within range}\} \geq \alpha$ is formulated as $1 - \prod r_j^{x_j} \geq \alpha$, which can be easily linearized. However, with correlated data this model is not valid anymore since the joint probability is not equal to the product of individual probabilities in this case. Assuming that the correlated data are moments up to level m , we develop a “distributionally robust” model as follows:

$$\begin{aligned}
& \max \sum_{i \in I} f_i y_i \\
& \sup_{\zeta \in \mathcal{P}^m} \left\{ \mathbb{P}^\zeta \left(\sum_{j \in J} \tilde{a}_{ij} x_j = 0 \right) \right\} \leq 1 - \alpha y_i \quad \forall i \in I \\
& \sum_{j=1}^{|J|} x_j = p \\
& y_i \in \{0, 1\} \quad \forall i \in I, \quad x_j \in \{0, 1\} \quad \forall j \in J,
\end{aligned}$$

where \tilde{a}_{ij} is a Bernoulli random number representing if sub-region i is covered by facility j . The above problem can be reformulated as the following approximation using FAM:

$$\begin{aligned}
& \max \sum_{i \in I} f_i y_i \\
& \max \{v_0^{(i)} : T_m v^{(i)} = S_m^{(i)}(x)\} \leq 1 - \alpha y_i \quad \forall i \in I \\
& \sum_{j=1}^{|J|} x_j = p \\
& y_i \in \{0, 1\} \quad \forall i \in I, \quad x_j \in \{0, 1\} \quad \forall j \in J, \quad v^{(i)} \in \mathbb{R}_+^{|J|+1} \quad \forall i \in I,
\end{aligned}$$

which is easy to linearize and formulate as a MIP.

Based on the independence assumption and the similar basic modeling technique, Daskin [42] developed models and algorithms for maximizing the expected demands that are satisfied given a limited number of facilities. Using FAM, our scheme can model this problem without any assumption on data independence.

$$\begin{aligned}
& \max \sum_{i \in I} f_i \left(\sum_{j \in J} v_j^{(i)} \cdot j \right) \\
& \text{s.t. } T_m v^{(i)} = S_m^{(i)}(x) \quad \forall i \in I \\
& \sum_{j=1}^{|J|} x_j = p \\
& x_j \in \{0, 1\} \quad \forall j \in J, \quad v^{(i)} \in \mathbb{R}_+^{|J|+1} \quad \forall i \in I,
\end{aligned}$$

where $v_j^{(i)}$ represents the probability that sub-region i is covered by j facilities. If $m = |J|$, then the above optimization problem produces an optimal allocation; otherwise, the optimal objective value provides an upper bound for the expected satisfied demands.

4.5.3 Probabilistic Shortest Path Problems

The problem is to route a vehicle on an s - t path through a network $G = (V, A)$, with $n = |A|$, where some arcs can be “blocked.” The travel time on arc $(i, j) \in A$ is c_{ij} if it is not blocked and $c_{ij} + M$ if it is blocked. Blocking is caused by another vehicle occupying the arc due to loading/unloading and has a fixed time $M > 0$. The goal is to find a path which is not too long and also to have an accurate prediction of the travel time on that path.

Assuming the probability distribution of blocking is given, a possible formulation

of the problem is as follows:

$$\begin{aligned}
& \min_{T,y} T \\
& \text{s.t.} \quad Ny = b, \quad y \in \{0,1\}^n \\
& \quad \Pr \left[\sum_{(ij) \in A} \tilde{c}_{ij} y_{ij} \leq T \right] \geq 1 - \epsilon.
\end{aligned} \tag{144}$$

In the above formulation, the first set of constraints are the usual shortest path constraints; \tilde{c}_{ij} is a random variable taking value c_{ij} with probability $1 - p_{ij}$ and $c_{ij} + M$ with probability p_{ij} (here p_{ij} is the probability that arc (i, j) is blocked); $1 - \epsilon \in (0, 1)$ is a desired confidence level; and T is the estimated shortest path length. Formulation (144) seeks a path and a minimum estimate T such that the path length is no bigger than T with high probability. In essence, the formulation minimizes the $(1 - \epsilon)$ -quantile of the path length distribution.

Let \tilde{a}_{ij} be a Bernoulli random variable that takes value 1 if arc (i, j) is blocked and 0 otherwise. Then $\tilde{c}_{ij} = c_{ij} + M\tilde{a}_{ij}$. The probabilistic constraint in (144) is equivalent to:

$$\Pr \left[\sum_{(ij) \in A} \tilde{c}_{ij} y_{ij} \leq T \right] \geq 1 - \epsilon \Leftrightarrow \Pr \left[\sum_{(ij) \in A} (c_{ij} + \tilde{a}_{ij}M) y_{ij} \leq T \right] \geq 1 - \epsilon$$

and can be written as:

$$\Pr \left[\sum_{(ij) \in A} \tilde{a}_{ij} y_{ij} \leq M^{-1} \left(T - \sum_{(ij) \in A} c_{ij} y_{ij} \right) \right] \geq 1 - \epsilon.$$

Note that if the actual travel time on the path given by y is T , then $M^{-1}(T - \sum_{(ij) \in A} c_{ij} y_{ij})$ is the number of blocked arcs on that path. Let us introduce binary variables z_i for $i \in \{0, 1, \dots, n\}$ such that z_i is 1 if there are i arcs blocked on the chosen path and zero otherwise. Then

$$\sum_{i=0}^n iMz_i \leq T - \sum_{(ij) \in A} c_{ij} y_{ij}, \quad \sum_{i=0}^n z_i = 1, \quad z_i \in \{0, 1\} \quad \forall i$$

models the relation between T , y , and z . The overall reformulation of (144) is then

$$\begin{aligned}
& \min_{T,y,z} T \\
& \text{s.t.} \quad Ny = b, \quad y \in \{0,1\}^n \\
& \quad \sum_{i=0}^n iMz_i \leq T - \sum_{(ij) \in A} c_{ij}y_{ij}, \quad \sum_{i=0}^n z_i = 1, \quad z_i \in \{0,1\} \quad \forall i \\
& \quad \Pr \left[\sum_{(ij) \in A} \tilde{a}_{ij}y_{ij} \leq \sum_{i=0}^n iz_i \right] \geq 1 - \epsilon.
\end{aligned} \tag{145}$$

Consider the event $B_{ij}(y_{ij}) = \{\tilde{a}_{ij}y_{ij} = 1\}$ for all $(i,j) \in A$. Then $\Pr[B_{ij}(y_{ij})] = p_{ij}y_{ij}$. The probabilistic constraint in formulation (145) requires that the probability that at most $\sum_{i=0}^n iz_i$ of the n events $B_{ij}(y_{ij})$ occur is at least $1 - \epsilon$.

For any $k \in \{1, \dots, n\}$, define

$$S_k(y) := \sum_{C \subseteq A, |C|=k} \Pr[\wedge_{(i,j) \in C} B_{ij}(y_{ij})].$$

Also let v_k for $k = 0, \dots, n$ be the probability that exactly k events occur. Then we have the following relationship

$$\sum_{k=0}^n v_k = 1, \quad \sum_{\ell=k}^n \binom{\ell}{k} v_\ell = S_k(y) \quad \forall k = 1, \dots, n, \quad v_k \geq 0 \quad \forall k = 0, 1, \dots, n.$$

The probability that at most $\sum_{i=0}^n iz_i$ of the events occur is then given by

$$\sum_{i=0}^n \left(\sum_{k=0}^i v_k \right) z_i.$$

We can now state a deterministic MINLP reformulation of (145):

$$\begin{aligned}
& \min_{T,y,z,v} T \\
& \text{s.t.} \quad Ny = b, \quad y \in \{0,1\}^n \\
& \quad \sum_{i=0}^n iMz_i \leq T - \sum_{(ij) \in A} c_{ij}y_{ij}, \quad \sum_{i=0}^n z_i = 1, \quad z_i \in \{0,1\} \quad \forall i \\
& \quad \sum_{i=0}^n \left(\sum_{k=0}^i v_k \right) z_i \geq 1 - \epsilon \\
& \quad \sum_{k=0}^n v_k = 1, \quad \sum_{\ell=k}^n \binom{\ell}{k} v_\ell = S_k(y) \quad \forall k = 1, \dots, n, \quad v_k \geq 0 \quad \forall k = 0, 1, \dots, n.
\end{aligned} \tag{146}$$

4.6 Conclusions

In this chapter, we introduced the probabilistic set k -cover problem and the associated ambiguous k -cover set. We developed a deterministic reformulation for the ambiguous k -cover set using the Boolean problem. The reformulation is exact but with exponential number of auxiliary variables. In order to build tractable approximations, we used linear program based bounds for the k -coverage probability, FAM and PAM, obtained by aggregating the Boolean problem. We also strengthened the probability bounds. The numerical results showed that the bound quality was significantly improved by the strengthened model. We built MIP formulations for the approximations based on FAM, PAM, and strengthened PAM. Computational experiments were conducted to demonstrate the quality of the deterministic reformulations in terms of cost effectiveness and solution robustness. The computational results also showed that the approximations based on the strengthened PAM provided better bounds than those based on FAM or PAM. At the end of the chapter, we demonstrated the flexibility of the modeling scheme developed in this chapter by formulating a number of applications involving Bernoulli random numbers.

There are several directions for future research under this topic. The first possibility is the investigation of solution approaches for solving the MIP formulations more efficiently. When n grows, the size of the MIP formulations will increase rapidly due to the auxiliary variables and the constraints used for linearization. Therefore, efficient solution algorithms or better formulations than the straightforward McCormick linearization are desirable. Second, the set k -cover structure and Bernoulli random data appear in many stochastic combinatorial optimization problems. Therefore, the second possible research direction could be finding more applications that can be handled by the models developed in this chapter.

REFERENCES

- [1] ABRAMS, Z., GOEL, A., and PLOTKIN, S., “Set k-cover algorithms for energy efficient monitoring in wireless sensor networks,” in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 424–432, ACM, 2004.
- [2] AGGARWAL, A., IMAI, H., KATOH, N., and SURI, S., “Finding k points with minimum diameter and related problems,” *Journal of algorithms*, vol. 12, no. 1, pp. 38–56, 1991.
- [3] AGRAWAL, S., DING, Y., SABERI, A., and YE, Y., “Price of correlations in stochastic optimization,” *Operations Research. Earlier version in SODA*, 2010.
- [4] AHMED, S., “Convexity and decomposition of mean-risk stochastic programs,” *Mathematical Programming*, vol. 106, no. 3, pp. 433–446, 2006.
- [5] AHMED, S. and PAPAGEORGIOU, D., “Probabilistic set covering with correlations,” *to appear in Operations Research*, 2013.
- [6] AKYILDIZ, I., SU, W., SANKARASUBRAMANIAM, Y., and CAYIRCI, E., “Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [7] ALON, N., AWERBUCH, B., and AZAR, Y., “The online set cover problem,” in *Proceedings of the 35th annual ACM Symposium on Theory of Computing*, pp. 100–105, ACM, 2003.
- [8] ALY, A. and WHITE, J., “Probabilistic formulation of the emergency service location problem,” *Journal of the Operational Research Society*, pp. 1167–1179, 1978.
- [9] AMALDI, E. and KANN, V., “The complexity and approximability of finding maximum feasible subsystems of linear relations,” *Theoretical Computer Science*, vol. 147, no. 1-2, pp. 181–210, 1995.
- [10] ANDREC, M., KHOLODENKO, B., LEVY, R., and SONTAG, E., “Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy,” *Journal of theoretical biology*, vol. 232, no. 3, pp. 427–441, 2005.
- [11] ATAMTÜRK, A. and NARAYANAN, V., “Polymatroids and mean-risk minimization in discrete optimization,” *Operations Research Letters*, vol. 36, no. 5, pp. 618–622, 2008.

- [12] ATAMTÜRK, A. and NARAYANAN, V., “The submodular knapsack polytope,” *Discrete Optimization*, vol. 6, no. 4, pp. 333–344, 2009.
- [13] ATAMTÜRK, A., NEMHAUSER, G. L., and SAVELSBERGH, M. W. P., “The mixed vertex packing problem,” *Mathematical Programming*, vol. 89, no. 1, Ser. A, pp. 35–53, 2000.
- [14] BALAS, E. and ZEMEL, E., “Critical cutsets of graphs and canonical facets of set-packing polytopes,” *Mathematics of Operations Research*, vol. 2, no. 1, pp. 15–19, 1977.
- [15] BALL, M., COLBOURN, C., and PROVAN, J., “Network reliability,” 1992.
- [16] BECKER, N. and STARCZAK, D., “Optimal vaccination strategies for a community of households,” *Mathematical Biosciences*, vol. 139, no. 2, pp. 117–132, 1997.
- [17] BEN-TAL, A. and NEMIROVSKI, A., “Robust solutions of linear programming problems contaminated with uncertain data,” *Mathematical Programming*, vol. 88, no. 3, pp. 411–424, 2000.
- [18] BERALDI, P. and RUSZCZYŃSKI, A., “The probabilistic set-covering problem,” *Oper. Res.*, vol. 50, no. 6, pp. 956–967, 2002.
- [19] BERALDI, P. and BRUNI, M. E., “An exact approach for solving integer problems under probabilistic constraints with random technology matrix,” *Ann. Oper. Res.*, vol. 177, pp. 127–137, 2010.
- [20] BERMAN, P. and DASGUPTA, B., “Approximating the online set multi-cover problems via randomized winnowing,” *Algorithms and Data Structures*, pp. 110–121, 2005.
- [21] BERMAN, P., DASGUPTA, B., and SONTAG, E., “Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks,” *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 39–50, 2004.
- [22] BERTSIMAS, D. and SIM, M., “The price of robustness,” *Operations research*, pp. 35–53, 2004.
- [23] BIENSTOCK, D., “Computational study of a family of mixed-integer quadratic programming problems,” *Mathematical Programming*, vol. 74, no. 2, pp. 121–140, 1996.
- [24] BONFERRONI, C. E., “Teoria statistica delle classi e calcolo delle probabilità,” *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, vol. 8, pp. 3–62, 1936.

- [25] BOOLE, G., “An investigation into the laws of thought,” *Walton and Maberly, London.[Reprinted (1951) Dover, New York.]*, 1854.
- [26] BOROS, E. and PREKOPA, A., “Closed form two-sided bounds for probabilities that at least r and exactly r out of n events occur,” *Mathematics of Operations Research*, pp. 317–342, 1989.
- [27] BROTCORNE, L., LAPORTE, G., and SEMET, F., “Ambulance location and relocation models,” *European journal of operational research*, vol. 147, no. 3, pp. 451–463, 2003.
- [28] CALAFIORE, G. and CAMPI, M. C., “Uncertain convex programs: randomized solutions and confidence levels,” *Math. Program.*, vol. 102, no. 1, Ser. A, pp. 25–46, 2005.
- [29] CALAFIORE, G. C. and EL GHAOU, L., “On distributionally robust chance-constrained linear programs,” *J. Optim. Theory Appl.*, vol. 130, no. 1, pp. 1–22, 2006.
- [30] CALAFIORE, G. and GHAOU, L., “On distributionally robust chance-constrained linear programs,” *Journal of Optimization Theory and Applications*, vol. 130, no. 1, pp. 1–22, 2006.
- [31] CARR, R., GREENBERG, H., HART, W., KONJEVOD, G., LAUER, E., LIN, H., MORRISON, T., and PHILLIPS, C., “Robust optimization of contaminant sensor placement for community water systems,” *Mathematical programming*, vol. 107, no. 1, pp. 337–356, 2006.
- [32] CHAN, T., “Low-dimensional linear programming with violations,” *SIAM Journal on Computing*, vol. 34(4), pp. 879–893, 2005.
- [33] CHARNES, A. and COOPER, W. W., “Deterministic equivalents for optimizing and satisficing under chance constraints,” *Operations Res.*, vol. 11, pp. 18–39, 1963.
- [34] CHARNES, A. and COOPER, W., “Chance-constrained programming,” *Management Science*, pp. 73–79, 1959.
- [35] CHEKURI, C., CLARKSON, K., and HAR-PELED, S., “On the set multi-cover problem in geometric settings,” in *Proceedings of the 25th annual symposium on Computational geometry*, pp. 341–350, ACM, 2009.
- [36] CHVÁTAL, V., “On certain polytopes associated with graphs,” *Journal of Combinatorial Theory B*, vol. 18, pp. 138–154, 1975.
- [37] CONFORTI, M. and WOLSEY, L., “Compact formulations as a union of polyhedra,” *Mathematical Programming*, vol. 114, no. 2, pp. 277–289, 2008.

- [38] CONSTANTINO, M., MILLER, A., and VAN VYVE, M., “Mixing MIR inequalities with two divisible coefficients,” *Mathematical Programming*, vol. 123, no. 2, pp. 451–483, 2010.
- [39] COPPERSMITH, D., LEE, J., LEUNG, J., and GÜNLÜK, O., *A polytope for a product of real linear functions in 0/1 variables*. IBM TJ Watson Research Center, 1999.
- [40] CRAMPIN, E., SCHNELL, S., and MCSHARRY, P., “Mathematical and computational techniques to deduce complex biochemical reaction mechanisms,” *Progress in biophysics and molecular biology*, vol. 86, no. 1, pp. 77–112, 2004.
- [41] DANTZIG, G. and INFANGER, G., “Multi-stage stochastic linear programs for portfolio optimization,” *Annals of Operations Research*, vol. 45, no. 1, pp. 59–76, 1993.
- [42] DASKIN, M., “A maximum expected covering location model: formulation, properties and heuristic solution,” *Transportation Science*, vol. 17, no. 1, pp. 48–70, 1983.
- [43] DASKIN, M. and STERN, E., “A hierarchical objective set covering model for emergency medical service vehicle deployment,” *Transportation Science*, vol. 15, no. 2, pp. 137–152, 1981.
- [44] DAWSON, D. A. and SANKOFF, D., “An inequality for probabilities,” *Proc. Amer. Math. Soc.*, vol. 18, pp. 504–507, 1967.
- [45] DE CAEN, D., “A lower bound on the probability of a union,” *Discrete mathematics*, vol. 169, no. 1, pp. 217–220, 1997.
- [46] DENTCHEVA, D., PRÉKOPA, A., and RUSZCZYŃSKI, A., “Bounds for probabilistic integer programming problems,” *Discrete Appl. Math.*, vol. 124, pp. 55–65, December 2002.
- [47] DENTCHEVA, D., PRÉKOPA, A., and RUSZCZYŃSKI, A., “Concavity and efficient points of discrete distributions in probabilistic programming,” *Math. Program.*, vol. 89, no. 1, Ser. A, pp. 55–77, 2000.
- [48] DEZA, M. and LAURENT, M., *Geometry of cuts and metrics*. Springer-Verlag, Berlin, 1996.
- [49] EDMONDS, J., *Combinatorial Structures and Their Applications*, pp. 69–87. New York: Gordon and Breach, 1970.
- [50] EFRAT, A., SHARIR, M., and ZIV, A., “Computing the smallest k-enclosing circle and related problems* 1,” *Computational Geometry*, vol. 4, no. 3, pp. 119–136, 1994.

- [51] ERDOĞAN, E. and IYENGAR, G., “Ambiguous chance constrained problems and robust optimization,” *Math. Program.*, vol. 107, no. 1-2, Ser. B, pp. 37–61, 2006.
- [52] ERDOĞAN, E. and IYENGAR, G., “Ambiguous chance constrained problems and robust optimization,” *Mathematical Programming*, vol. 107, no. 1, pp. 37–61, 2006.
- [53] ERDOĞAN, E. and IYENGAR, G., “On two-stage convex chance constrained problems,” *Math. Methods Oper. Res.*, vol. 65, no. 1, pp. 115–140, 2007.
- [54] FEIGE, U., “A threshold of $\ln n$ for approximating set cover,” *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.
- [55] GALAMBOS, J., “Bonferroni inequalities,” *The Annals of Probability*, pp. 577–581, 1977.
- [56] GAO, S., VU, C., and LI, Y., “Sensor scheduling for k-coverage in wireless sensor networks,” *Mobile Ad-hoc and Sensor Networks*, pp. 268–280, 2006.
- [57] GAREY, M. and JOHNSON, D., *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman & Co. New York, NY, USA, 1979.
- [58] GENDREAU, M., LAPORTE, G., and SEMET, F., “Solving an ambulance location model by tabu search,” *Location science*, vol. 5, no. 2, pp. 75–88, 1997.
- [59] GHOSH, A. and DAS, S., “Coverage and connectivity issues in wireless sensor networks: A survey,” *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.
- [60] GLOVER, F. and WOOLSEY, E., “Converting the 0-1 polynomial programming problem to a 0-1 linear program,” *Operations Research*, pp. 180–182, 1974.
- [61] GOEMANS, M. and VONDRÁK, J., “Stochastic covering and adaptivity,” in *LATIN 2006: Theoretical informatics*, vol. 3887 of *Lecture Notes in Comput. Sci.*, pp. 532–543, Berlin: Springer, 2006.
- [62] GUAN, Y., AHMED, S., and NEMHAUSER, G., “Sequential pairing of mixed integer inequalities,” *Discrete Optimization*, vol. 4, no. 1, pp. 21–39, 2007.
- [63] GÜNLÜK, O. and POCHET, Y., “Mixing mixed integer inequalities,” *Mathematical Programming*, vol. 90, no. 3, pp. 429–457, 2001.
- [64] GURVICH, I., LUEDTKE, J., and TEZCAN, T., “Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach,” *Management Science*, vol. 56, no. 7, pp. 1093–1115, 2010.
- [65] HAIGHT, R., REVELLE, C., and SNYDER, S., “An integer optimization approach to a probabilistic reserve site selection problem,” *Operations Research*, pp. 697–708, 2000.

- [66] HAILPERIN, T., “Best possible inequalities for the probability of a logical function of events,” *The American Mathematical Monthly*, vol. 72, no. 4, pp. 343–359, 1965.
- [67] HALL, D. and LLINAS, J., *Handbook of multisensor data fusion*. CRC Press, 2001.
- [68] HEFEEDA, M. and AHMADI, H., “A probabilistic coverage protocol for wireless sensor networks,” in *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*, pp. 41–50, IEEE, 2007.
- [69] HOGAN, K. and REVELLE, C., “Concepts and applications of backup coverage,” *Management Science*, pp. 1434–1444, 1986.
- [70] HOLDER, A., ed., *Mathematical Programming Glossary*. <http://glossary.computing.society.informs.org>: INFORMS Computing Society, 2006–08. Originally authored by Harvey J. Greenberg, 1999–2006.
- [71] HUANG, C. and TSENG, Y., “The coverage problem in a wireless sensor network,” *Mobile Networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.
- [72] JOHNSON, D., “Approximation algorithms for combinatorial problems*,” *Journal of Computer and System Sciences*, vol. 9, no. 3, pp. 256–278, 1974.
- [73] KHOLODENKO, B., KIYATKIN, A., BRUGGEMAN, F., SONTAG, E., WESTERHOFF, H., and HOEK, J., “Untangling the wires: a strategy to trace functional interactions in signaling and gene networks,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 20, p. 12841, 2002.
- [74] KOCH, T., ACHTERBERG, T., ANDERSEN, E., BASTERT, O., BERTHOLD, T., BIXBY, R. E., DANNA, E., GAMRATH, G., GLEIXNER, A. M., HEINZ, S., LODI, A., MITTELMANN, H., RALPHS, T., SALVAGNIN, D., STEFFY, D. E., and WOLTER, K., “MIPLIB 2010,” *Mathematical Programming Computation*, vol. 3, no. 2, pp. 103–163, 2011.
- [75] KUAI, H., ALAJAJI, F., and TAKAHARA, G., “A lower bound on the probability of a finite union of events,” *Discrete Mathematics*, vol. 215, no. 1-3, pp. 147–158, 2000.
- [76] KÜÇÜKYAVUZ, S., “On mixing sets arising in chance-constrained programming,” *Mathematical Programming*, vol. 132, no. 1-2, pp. 31–56, 2012.
- [77] KUMAR, S., LAI, T., and BALOGH, J., “On k-coverage in a mostly sleeping sensor network,” in *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 144–158, ACM, 2004.
- [78] KWEREL, S. M., “Bounds on the probability of the union and intersection of m events,” *Advances in Appl. Probability*, vol. 7, pp. 431–448, 1975.

- [79] KWEREL, S. M., “Most stringent bounds on the probability of the union and intersection of m events for systems partially specified by $S_1, S_2, \dots, S_k, 2 \leq k < m$,” *J. Appl. Probability*, vol. 12, no. 3, pp. 612–619, 1975.
- [80] LO, C., MATOUŠEK, J., and STEIGER, W., “Algorithms for ham-sandwich cuts,” *Discrete and Computational Geometry*, vol. 11, no. 1, pp. 433–452, 1994.
- [81] LUEDTKE, J. and AHMED, S., “A sample approximation approach for optimization with probabilistic constraints,” *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.
- [82] LUEDTKE, J., AHMED, S., and NEMHAUSER, G. L., “An integer programming approach for linear programs with probabilistic constraints,” *Mathematical Programming*, vol. 122, no. 2, pp. 247–272, 2010.
- [83] LUEDTKE, J., “An integer programming and decomposition approach to general chance-constrained mathematical programs,” in *Integer Programming and Combinatorial Optimization* (EISENBRAND, F. and SHEPHERD, F., eds.), vol. 6080 of *Lecture Notes in Computer Science*, pp. 271–284, Springer Berlin / Heidelberg, 2010.
- [84] LUEDTKE, J., “A branch-and-cut decomposition algorithm for solving general chance-constrained mathematical programs,” *Computer Sciences Technical Report*, May, 2011.
- [85] MAGNANTI, T. and RAGHAVAN, S., “Strong formulations for network design problems with connectivity requirements,” *Networks*, vol. 45, no. 2, pp. 61–79, 2005.
- [86] MARKOWITZ HARRY, M., “Portfolio selection,” *Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [87] MATOUŠEK, J., “On geometric optimization with few violated constraints,” *Discrete and Computational Geometry*, vol. 14, no. 1, pp. 365–384, 1995.
- [88] MCCORMICK, G., “Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems,” *Mathematical Programming*, vol. 10, no. 1, pp. 147–175, 1976.
- [89] MCCORMICK, G., “Computability of global solutions to factorable nonconvex programs: Part iconvex underestimating problems,” *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.
- [90] MEGIDDO, N., “Linear-time algorithms for linear programming in r_3 and related problems,” in — *23rd Annual Symposium on Foundations of Computer Science*, pp. 329–338, IEEE, 1982.

- [91] MEHTA, D., LOPEZ, M., and LIN, L., “Optimal coverage paths in ad-hoc sensor networks,” in *Communications, 2003. ICC’03. IEEE International Conference on*, vol. 1, pp. 507–511, IEEE, 2003.
- [92] MILLER, A. and WOLSEY, L., “Tight formulations for some simple mixed integer programs and convex objective integer programs,” *Mathematical Programming*, vol. 98, no. 1, pp. 73–88, 2003.
- [93] NEMHAUSER, G. and WOLSEY, L., *Integer and combinatorial optimization*, vol. 18. Wiley New York, 1988.
- [94] NEMIROVSKI, A. and SHAPIRO, A., “Scenario approximations of chance constraints,” *Probabilistic and randomized methods for design under uncertainty*, pp. 3–47, 2006.
- [95] PAGNONCELLI, B., AHMED, S., and SHAPIRO, A., “Computational study of a chance constrained portfolio selection problem,” *Journal of Optimization Theory and Applications*, vol. 142, pp. 399–416, 2009.
- [96] PAPAGEORGIOU, D. and HOLENDER, M., “Track-to-track association and ambiguity management in the presence of sensor bias,” in *Information Fusion, 2009. FUSION’09. 12th International Conference on*, pp. 2012–2019, IEEE, 2009.
- [97] PRÉKOPA, A., “On probabilistic constrained programming,” in *Proceedings of the Princeton symposium on mathematical programming*, pp. 113–138, Princeton, New Jersey: Princeton University Press, 1970.
- [98] PRÉKOPA, A., “Boole-bonferroni inequalities and linear programming,” *Operations Research*, pp. 145–162, 1988.
- [99] PRÉKOPA, A., “Sharp bounds on probabilities using linear programming,” *Operations Research*, pp. 227–239, 1990.
- [100] PRÉKOPA, A. and GAO, L., “Bounding the probability of the union of events by aggregation and disaggregation in linear programs,” *Discrete applied mathematics*, vol. 145, no. 3, pp. 444–454, 2005.
- [101] REVELLE, C. and HOGAN, K., “The maximum availability location problem,” *Transportation Science*, vol. 23, no. 3, pp. 192–200, 1989.
- [102] ROCKAFELLAR, R. T., “Optimization under uncertainty- lecture notes,” 2001.
- [103] ROSSI, F., SMRIGLIO, S., and SASSANO, A., “Models and algorithms for terrestrial digital broadcasting,” *Annals of Operations Research*, vol. 107, no. 1, pp. 267–283, 2001.
- [104] RUSZCZYŃSKI, A. and SHAPIRO, A., “Stochastic programming,” *Handbooks in operations research and management science*, vol. 10, 2003.

- [105] SASSANO, A., “On the facial structure of the set covering polytope,” *Mathematical Programming*, vol. 44, pp. 181–202, 1989.
- [106] SATHE, Y., PRADHAN, M., and SHAH, S., “Inequalities for the probability of the occurrence of at least m out of n events,” *Journal of Applied Probability*, pp. 1127–1132, 1980.
- [107] SAXENA, A., GOYAL, V., and LEJEUNE, M. A., “MIP reformulations of the probabilistic set covering problem,” *Math. Program.*, vol. 121, no. 1, Ser. A, pp. 1–31, 2010.
- [108] SHAPIRO, A., DENTCHEVA, D., and RUSZCZYŃSKI, A., *Lectures on stochastic programming: modeling and theory*, vol. 9. Society for Industrial Mathematics, 2009.
- [109] SHAPIRO, A. and HOMEM-DE MELLO, T., “On the rate of convergence of optimal solutions of monte carlo approximations of stochastic programs,” *SIAM journal on optimization*, vol. 11, no. 1, pp. 70–86, 2000.
- [110] SHEN, S., SMITH, J., and AHMED, S., “Expectation and chance-constrained models and algorithms for insuring critical paths,” *Management Science*, vol. 56, no. 10, p. 1794, 2010.
- [111] SONG, Y. and LUEDTKE, J., “Branch-and-cut approaches for chance-constrained formulations of reliable network design problems,” 2011.
- [112] SONTAG, E., KIYATKIN, A., and KHOLODENKO, B., “Inferring dynamic architecture of cellular networks using time series of gene expression, protein and metabolite data,” *Bioinformatics*, vol. 20, no. 12, p. 1877, 2004.
- [113] TAKACS, L., “On the method of inclusion and exclusion,” *Journal of the American Statistical Association*, pp. 102–113, 1967.
- [114] TANNER, M. and NTAIMO, L., “IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation,” *European Journal of Operational Research*, vol. 207, no. 1, pp. 290–296, 2010.
- [115] THOMAS, R. and WIDMAYER, P., “ k -violation linear programming,” *Information Processing Letters*, vol. 52, pp. 109–114, 1994.
- [116] TUNCEL, A., PRECIADO, F., RARDIN, R., LANGER, M., and RICHARD, J.-P., “Strong valid inequalities for fluence map optimization problem under dose-volume restrictions,” *Annals of Operations Research*, pp. 1–22.
- [117] VAZIRANI, V., *Approximation algorithms*. Springer Verlag, 2001.
- [118] YONGJIA SONG, SIMGE KUCUKYAVUZ, J. L., “A branch-and-cut algorithms for the chance constrained knapsack problem,” *Technique report*, 2012.

- [119] ZHAO, M. and DE FARIAS JR, I., “The mixing-MIR set with divisible capacities,” *Mathematical Programming*, vol. 115, no. 1, pp. 73–103, 2008.
- [120] ZOU, Y. and CHAKRABARTY, K., “Sensor deployment and target localization in distributed sensor networks,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61–91, 2004.